IoT Threat Mitigation Engine empowered by Artificial Intelligence Multi-Objective Optimization*

Asterios Mpatziakas[a],*, Anastasios Drosou[a], Stavros Papadopoulos[a] and Dimitiris Tzovaras[a]

[a]Information Technologies Institute, Centre of Research & Technology - Hellas 6th km Harilaou - Thermi, 57001, Thermi - Thessaloniki, Greece

ABSTRACT

From smart homes to smart cities and Industry 4.0 to Transportation Systems, Internet of Things (IoT) is a domain which promises incredible growth coupled with great impact, in numerous fields. IoT networks are composed of numerous different Things, arranged in diverse topologies with diverse needs. This diversity is partially due to the numerous areas where IoT applications are utilized, which at their entirety can be referred as the IoT ecosystem. The IoT ecosystem suffers from numerous vulnerabilities, due to reasons such as design flows, hardware limitation or simply human error and is subject to various attacks targeting IoT services, platforms and networks. These attacks can have significant consequences such as economic losses, service disruption or data leaks. Cyber-attacks are an unavoidable and must be faced in tandem with the global growth of IoT networks. An approach that can assist in developing robust, intelligent Cyber-security tools for IoT is using Artificial Intelligence. In the following paper, a mechanism is presented that automatically selects appropriate mitigation actions in an optimal way to countermeasure attacks faced by IoT networks. This is achieved by using an novel Artificial Intelligence mechanism based on a Deep Neural Architecture called Pointer Networks to optimize security-related KPIs. Experimental results, show that the proposed method produces equal or better Pareto optimal solutions, performs faster compared to state-of-the-art (SoA) algorithms and scales better.

1. INTRODUCTION

The number of devices with internet connectivity, has been rapidly increasing with some estimations claiming that approximately 50 billion Internet of Things (IoT) devices are connected by the end of 2020 (Tahsien et al., 2020) . Internet of Things (IoT) networks can be composed of numerous different "Things" e.g. sensors, gateways, protocols, network or application servers applied in use cases such as healthcare, transportation systems, cities, supply chains, environmental monitoring, manufacturing etc. which at their entirety can be referred as the IoT ecosystem.

Along with the expansion of the IoT devices and the related network infrastructures, came an increase of security related risks and issues, threatening the continuity of provided services and posing privacy dangers in terms of data availability, integrity and confidentiality both at the user or infrastructure or service provider level. The application of standard cyber-security methodologies can be challenging due to the heterogeneity of IoT networks (Pal et al., 2020). Additionally, IoT ecosystems with time sensitive services have emerged e.g., intelligent transportation systems, smart manufacturing or medical care (Khadr et al., 2019).

*This work is supported by the European Unions' Horizon 2020 Research and Innovation Program through the SerIoT project under Grant Agreement No. 780139. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission.

ampatziakas@iti.gr (A. Mpatziakas); drosou@iti.gr (A. Drosou); spap@iti.gr (S. Papadopoulos); dimitrios.tzovaras@iti.gr (D. Tzovaras)

ORCID(s): 0000-0003-4385-9491 (A. Mpatziakas)

In this context, cyber-attacks need to be detected in a timely manner allowing for prompt mitigation, since the realization of a security failure can cause severe consequences ranging from monetary damages to physical harm. Implementing effective secure IoT platforms and networks has value for both the industry and the end-users.

To combat the emerging plethora of the various attacks targeting the dynamic environment of IoT networks and devices, it is critical to develop novel, smart tools that can handle the diverse nature of such attacks and block or counter them. A solid basis for such tools can be Artificial Intelligence (AI) based methods (Zhang and Tao, 2020; Yan et al., 2020).

In the last decade the AI field has experienced explosive growth, partially due to Deep Learning (DL) and Deep Neural Networks (DNN) with ground-breaking applications in multiple domains. Combining AI and IoT networks and devices, is a subject that has attracted the attention of both academic and private research interests (Zhang and Tao, 2020), while the application of AI algorithms to the IoT domain has been described as Artificial Intelligence-of Things (AIoT) (Lai et al., 2021). A few examples of the successful application of Deep Learning based AI methods in the IoT domain include either use-case or device specific applications such as wearable devices (Ravì et al., 2017), patient rehabilitation systems (Fan et al., 2014) and smart energy meters (Alahakoon and Yu, 2016). AI methods have also been used in applications that manage and affect the functionality of the IoT network. Examples of this category include using AI for routing protocol handling (Al-Janabi

and Al-Raweshidy, 2018), Data placement oriented towards privacy preservation Xu et al. (2018), to empower a Anomaly Detection module (Protogerou et al., 2020), malware detection Toldinas et al. (2021), intrusion detection Awan et al. (2021), threat analysis Ali et al. (2022) or for the resource allocation handling of an IoT network (Ramírez et al., 2020). Apart from Deep Learning based AI, another category of AI algorithms that have shown robust performance in optimization problems related to IoT are nature-inspired algorithms. Their applications span from the optimal scaling of the architecture of an IoT network (Hildmann et al., 2019) to energy-efficient routing control (Khan et al., 2020) or the verification of the security of the runtime and routing policies of an SDN empowered IoT network (Papachristou et al., 2019). In (Kaul et al., 2021), an extensive review of the application on such algorithms in various computing systems including IoT networks is provided.

Approaches using AI, can be combined with Software defined networking (SDN) technologies. SDN based architectures offer programmable networks that separate the traffic plane from the control plane (Bera et al., 2017), allowing control of the various network layers through the network flows (Akyildizetal.,2014). Coupling AI with SDN technologies can enhance the SDN capabilities, leading to better decision making (Latah and Toker, 2019) and more robust security solutions Correa Chica et al. (2020).

Within this context, an extendable mechanism for the unsupervised selection of mitigation actions in IoT network is presented. This module uses an original AI based methodology that employs deep neural networks called Pointer Networks (Vinyals et al., 2015) to decide optimal strategies against threats by solving multi-objective problems to optimize the value of cyber-security KPIs. The mechanism is built to interact with SDN systems in order to enforce the mitigation strategies to the network.

1.1. Key contributions

In the following sections, we present an approach using novel multi-objective (MO) method, based on a Deep Neural architecture called Pointer Networks (Vinyals et al., 2015) (PointerNet), for optimizing the selection of countermeasures of an IoT network under attack. While countermeasure selection using MO techniques is well studied in 'classic' computer networks, it is a fairly new topic in the context of IoT. To the best of our knowledge, both an MO approach using Pointer Networks and application of Pointer Networks for IoT security and have not been yet studied in the literature.

The problem under investigation can be stated as follows: Assume X devices of different types are under threat or an attack, each with n mitigation actions, unique to its type, to counter the threat or attack. Select one appropriate mitigation action for each device while optimizing a number of metrics and respecting a number of constrains. This corresponds to a class of problems, known as MO 0/1 Knapsack Problems which is known to be NP-Hard. The Knapsack problem belongs to the category of Discrete Combinatorial (DC) problems and is met in numerous fields: a class of algorithms known as Evolutionary Algorithms is considered the standard in solving them (Liu et al., 2020; Emmerich and Deutz, 2018).

Pointer networks is a Deep Neural Architecture using Reinforcement Learning, that was first proposed by (Vinyals et al., 2015) to solve the Traveling Salesman Problem which also belongs to the DC problem class. Variations of Pointer networks were proposed to solve the Single-Objective (SO) 0/1 knapsack problem in (Bello et al., 2017; Gu and Hao, 2018).

In this paper, a novel approach is proposed, to solve the MO version of 0/1 knapsack problems resulting in a method that produces equal or better Pareto optimal solutions, performs faster compared to state-of-the-art (SoA) algorithms and scales better. This is shown by experimental results presented in section 4.2. The suggested process takes into account multiple aspects of the network and results in a set of optimal trade-off solutions between multiple cyber-security KPIs, a non-trivial problem since there is no single solution that simultaneously optimizes each KPI. The general position of the proposed Mitigation Engine module in an IoT network is shown in figure 1: The Mitigation Engine receives Anomalies from an Intrusion Detection System and outputs its' decisions to be applied to the network via the SDN controller.

The remainder of the paper is structured as follows: Section 2 contains a survey of the relevant review and Section 3 presents the methodology used to formulate the proposed approach while in Section 4 experimental results for the proposed Mitigation Mechanism are presented. Finally, Section 5 concludes the articles and describes future work directions.

2. Related Work

The following section presents the results of a literature survey related to the proposed approach: Initially, a brief literature review about threat mitigation in IoT is presented. Then, the subject of multi-objective (MO) optimization problems in the IoT domain is examined along with a cursory introduction on common approaches to solve them. The final part of this section contains the results of an extensive literature review on Key Performance Indicators (KPI) for IoT Cyber-security. Based

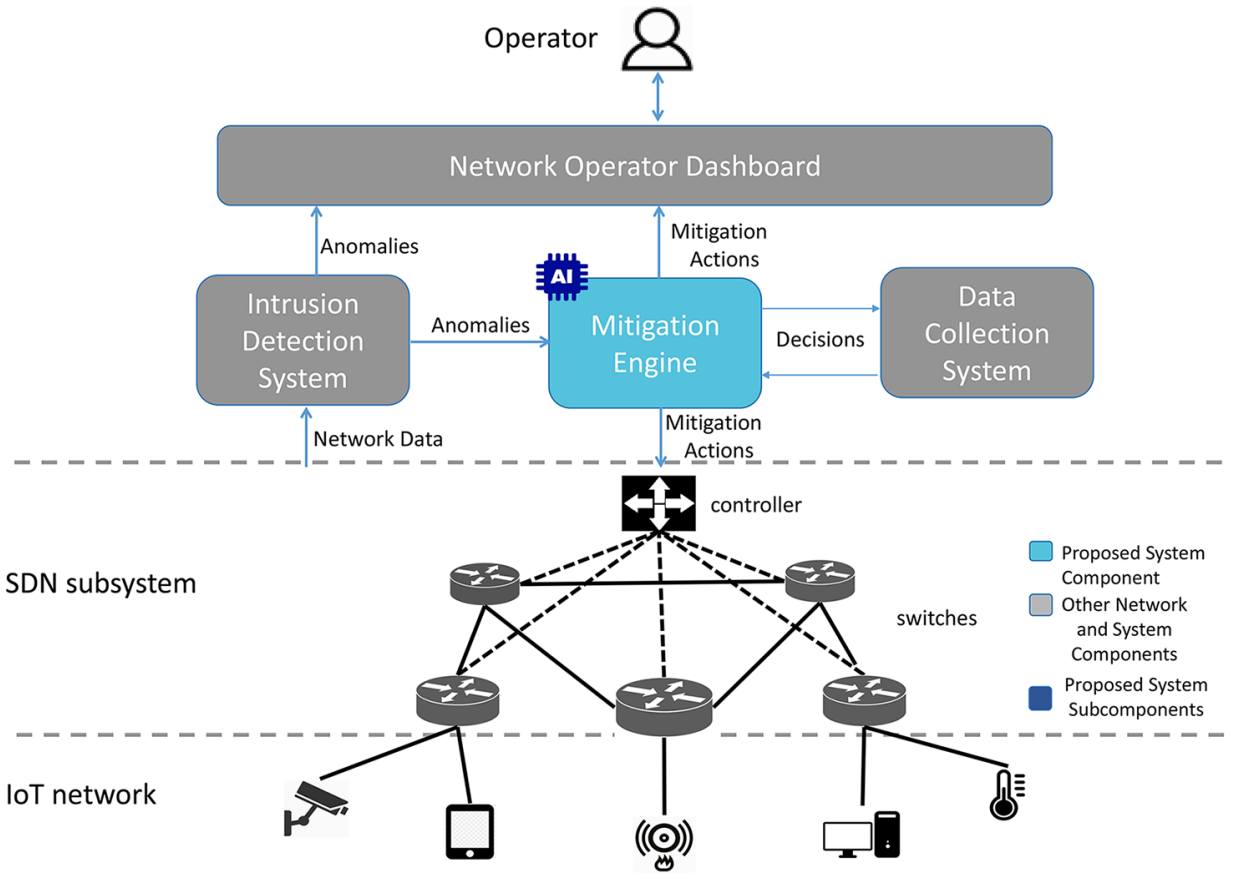


Fig. 1: High level overview of the position of the proposed Mitigation Engine module in an IoT network.

on this survey, the KPIs that are used as targets for the optimization process are decided.

2.1. Attack and Threat mitigation in IoT networks

There are two distinct alternatives to mitigations of threats in cyber-systems under attack, including IoT networks: The first deals with the mitigation of a single attack type or method e.g., a sinkhole attack where the attacker causes large traffic flows to pass from a node he controls to obtain data, can be mitigated by more secure protocols. Numerous studies contain comprehensive reviews of such approaches, for example (Ahemd et al., 2017; Mahmoud et al., 2015).

The second alternative is to use a scheme to select the appropriate countermeasures to the threats or attacks faced by the system based on the values of some Key Performance Indicators. This approach offers a holistic treatment of threat mitigation in the sense that it allows the selection of countermeasures for attacks from multiple sources, with multiple possible steps. This approach was chosen to be followed in the presented Mitigation Engine implementation.

We distinguished solutions falling under the second category further, to four separate classes: The first class includes approaches with no automation that measure the values of one or more KPIs. However, no optimization methods are applied and no suggestions or automated responses for the appropriate mitigation strategy are offered. An example, is the framework suggested by Gonzalez-Granadillo et. al (Gonzalez-Granadillo et al., 2015), in which the values for some KPIs, corresponding to multiple sets of different mitigation action choices, are presented numerically and visually to the Security operator of the system and then she must reach a decision on which actions to deploy.

The second class involves the automated mitigation of attacks using Heuristic methods based on thresholds for the values of KPIs. Based on predefined scenarios and values, the system chooses from a list of predefined actions to react to the threats. This approach can lead to complex patterns of different cases and values that can be difficult to scale up. An example of such an approach can be found in (Kotenko and Doynikova, 2016).

The third class includes approaches where the selection of mitigation actions is based on the optimization of the value a single KPI or transforming the problem to a single objective (SO) problem e.g. (Chehida et al., 2020) where the authors select countermeasures based on minimizing the cost to deploy them.

Finally, the fourth class includes the approaches where the selection of mitigation actions is based on the optimization of the values of multiple KPIs which might be antagonistic to each other but better describe the impact

Table 1
Literature review results on the selection of mitigation actions for cyber-attacks

Approach Category	Approaches
No automation	N=3, Visualization (Gonzalez-Granadillo et al., 2015; Granadillo et al., 2012; Gonzalez-Granadillo et al., 2017)
Heuristic Approach	N=3, (Chung et al., 2013; Kotenko and Doynikova, 2016; Doynikova and Kotenko, 2018)
Single-objective Optimization	N=8, Ant Colony (Wang et al., 2013), Bellman Method (Zonouz et al., 2009; Zonouz and Haghani, 2013; Miehling et al., 2015), Genetic Algorithm (Poolsappasit et al., 2012; Garzia et al., 2017; Li et al., 2018; Chehida et al., 2020)
Multi-objective Optimization	N=10, Branch & Bound Integer Optimization (Roy et al., 2012),Genetic Algorithm (Poolsappasit et al., 2012), NSGA-II algorithm (Dewri et al., 2012; Rachedi and Benslimane, 2016; Lee et al., 2017; Hasan et al., 2018; Enoch et al., 2019), Single Additive Weighting & Weighted Product method (Shameli-Sendi and Dagenais, 2015; Shameli-Sendi et al., 2018), Tabu Search algorithm (Viduto et al., 2012)

of the action on the system. This approach was chosen for the proposed Countermeasure selection module. Table 1 presents the results of the literature review performed along with details on the methods used.

2.2. Multi-objective optimization in the context of IoT networks and approaches to solve such problems

Optimizing the solution of problems based on multiple objectives (MO) has been effectively applied in many diverse fields such as economics, multi-modal analysis, or engineering (Kalamaras et al., 2014). Additionally, it has been used to efficiently solve various IoT related problems such as IoT network design (Huang et al., 2015), IoT wireless network spectrum allocation (Han et al., 2018) or IoT quality of service handling (Prasanth and Jayachitra, 2020).

MO approaches have been used to select a number of countermeasures from a predefined set of such actions in the context of cyber-security as shown in Table 1. In the context of IoT networks specifically, KPI optimization for mitigation action selection has become a recent topic of interest. Three solutions involving SO approaches for mitigation selection have been identified, either in generic IoT networks (Chehida et al., 2020; Garzia et al., 2017) or smart energy grid systems (Zonouz and Haghani, 2013). Two IoT specific studies using MO have been identified i.e. (Rachedi and Benslimane, 2016) where the authors optimize multiple KPIs to secure a Sensor Network while in (Hasan et al., 2018) MO is used in an industrial energy delivery system based on IoT.

Another manner to solve the problem of mitigation action selection is using tools from game or auction theory, e.g., the approach used in (Rontidis et al., 2015). However, while related to MO optimization, these tools assume prior expert knowledge about the specific workings of the systems such as user payoffs or user/attacker profit and loss margins and more important they assume competing agents that are linked to different objectives thus are not considered to be in the scope of this paper.

An approach commonly used to solve MO problems, is the utilization of the so-called decomposition techniques such as scalarization where the multiple objectives are transformed into a single objective problem, usually using arbitrarily decided weights to account for the importance of each objective function.

Another approach, is to use so called Evolutionary or Nature Inspired Algorithms which are subset of evolutionary computation. Along with techniques and methods such as fuzzy logic, swarm intelligence or artificial neural networks, Evolutionary algorithms belong to Computational Intelligence field of study, also known as soft computation, a subdiscipline of the Artificial Intelligence field (Back et al., 1997).

Evolutionary Algorithms share the following common characteristics: they can effectively represent numerical knowledge, easily adapt to changes of the input data while efficiently producing solutions in computationally hard problems (Siddique and Adeli, 2013). In general, these algorithms evolve a population of candidate solutions towards an optimized solution space. The fitness of each individual belonging to the population, is computed through use of problem specific objective functions. The individuals with the highest function scores are used as the basis of a new generation of the population. However, it is known that Evolutionary algorithms suffer when scaling up to larger problems is required (Sloss and Gustafson, 2020).

In MO problems, optimizing the value of a single KPI can result in undesirable effects with respect to the values of the other KPIs of interest. Instead, the desired solutions need to satisfy the KPI objectives without being dominated by other solutions, i.e., solutions where none of the values

Table 2
KPIs used in literature to measure mitigation and countermeasure efficiency for cyber-security

KPI Category/Name	*Number of appearances in literature and other names used*
Positive impact of the countermeasures chosen on the security of the system	N = 21 : Positive Impact (Shameli-Sendi and Dagenais, 2015), Benefit (Chung et al., 2013), Security Quality (Shameli-Sendi et al., 2018), Stateless Security (Enoch et al., 2019), Security Benefit (Li et al., 2018), Security level (Zonouz and Haghani, 2013; Hasan et al., 2018), Total Initial Risk (Viduto et al., 2012), Security Risk (Garzia et al., 2017), Vulnerability Surface Coverage (Shameli-Sendi et al., 2018), Weakness Coverage (Lee et al., 2017)
Negative impact of the countermeasures chosen on the security of the system	N = 9 : Residual Damage (Dewri et al., 2012), Negative Impact (Shameli-Sendi and Dagenais, 2015), Intrusiveness (Chung et al., 2013), Impact (Roy et al., 2012), Security Performance (Shameli-Sendi et al., 2018), Negative Impact on QoS (Li et al., 2018), System Downtime(Wang et al., 2013), CVSS score (Wang et al., 2013; Chung et al., 2013)
Cost to prepare and deploy selected countermeasures	N = 8 : Service Cost (Shameli-Sendi et al., 2018), Cost (Enoch et al., 2019; Wang et al., 2013; Chung et al., 2013; Miehling et al., 2015), Security Control Cost (Dewri et al., 2012; Poolsappasit et al., 2012) , Deployment Cost (Li et al., 2018)
Monetary profit gained / Loss averted by deploying selected countermeasures	N = 8 : ROI (Chung et al., 2013), RORI (Kheir et al., 2010; Granadillo et al., 2012), Loss/Gains(Poolsappasit et al., 2012), Benefit of implementation (Doynikova and Kotenko, 2018), Recovery Cost (Zonouz and Haghani, 2013), Total Investment (Viduto et al., 2012), Monetal Cost (19)
Hypervolume of users, channels, resources	N = 3 : (Garcia-Alfaro, 2017; Gonzalez-Granadillo et al., 2017, 2015)
System Throughput	N = 2 : (Hasan et al., 2018; Varga et al., 2018)
E2E latency	N = 1: (Hasan et al., 2018)
Detection Time	N = 1 : (Varga et al., 2018)

of the objective functions examined can be improved without the value of some other objective deteriorating. Solutions with this property are called non-dominated or Pareto optimal. The final result is a set of optimal trade-off solutions between multiple KPIs, often referred to as the Pareto Front (Emmerich and Deutz, 2018).

2.3. Key performance Indicators for IoT Cyber-security countermeasures

To assess the Key Performance Indicators most commonly used to measure the efficiency of mitigation actions and countermeasures, an extensive literature review was performed. Based on it, as shown in Table 2, four major KPI categories where distinguished:

- KPIs measuring the "Positive impact of a countermeasure chosen on the security of the system" e.g., in terms of risk of an attack averted or using a custom-defined Security Level.
- KPIs measuring the "Negative impact of a countermeasure chosen on the security of the system" e.g., the down time that might be caused by a countermeasure for some parts of the system or the possible residual damage despite the application of a mitigation action.
- KPIs measuring the "Cost to prepare and deploy selected countermeasures" in terms that might include but are not strictly monetary, e.g., the cost that occurs from the system resources consumed when a mitigation action is applied.
- KPIs measuring the "Monetary profit gained or Loss averted by deploying selected countermeasures" which are strictly measured in monetary units or percentage of monetary gain/loss.

2.4. Security Related Key Performance Indicators for the selection of Mitigation Actions

The following section presents the four Key Performance Indicators (KPIs) selected to be used in this report, as the metrics to be optimized, towards choosing which mitigation actions should be deployed to secure the network. The choice was made based on the review presented in Section 2.1.

2.4.1Common Vulnerability Scoring System

Common Vulnerability Scoring System (CVSS) is an open Industry standard for assessing the severity of a cybersecurity vulnerability (Mell et al., 2006). A vulnerability has a CVSS score ∈ [0,10] with 10 representing the highest severity. For the calculation of CVSS scores predefined values and equations are used. These are presented in the white-paper "Vulnerability Scoring System version 3.1 Specification Document" (Group, 2019): Each vulnerability is described by two strictly defined sets of metrics. The first is the set of **Exploitability** metrics which reflect the characteristics of the vulnerable device or component. These include:

1. Attack Vector which describes the context in which a vulnerability is exploited (Network, Adjacent Network, Local or Physical),
2. Attack Complexity which describes the existence of any special conditions required (or lack of them) for a vulnerability to be exploited (Low, High),
3. Privileges Required (or lack of them) describes the level of privileges required by the attacker for a vulnerability to be exploited (None, Required),
4. User Interaction describes the existence of any user other than the attacker required (or lack of them) for a vulnerability to be exploited (None, Required),
5. Scope, i.e. examining if impacts caused to systems beyond the vulnerable component (Unchanged, Changed)

and the second is the set of **Impact Metrics** which reflect the effects of a successful exploitation of a vulnerability:

1. Confidentiality reflects the impact to the confidentiality of the data managed by the device or application affected by the vulnerability to be exploited (None, Low, High),
2. Integrity reflects the impact to the integrity, i.e. trustworthiness and veracity, of the data managed by the device or application affected by the vulnerability to be exploited (None, Low, High) and
3. Availability reflects the impact to the availability of the device or application affected by the vulnerability to be exploited (None, Low, High) The actual formulas and values are omitted for the sake of brevity but can be found in (Group, 2019). Using CVSS, security experts can easily share discovered vulnerabilities via public databases such as the National Vulnerability Database (Booth et al., 2013). Moreover, any attack can be translated to a vulnerability. In Section 4 the following formula is used to calculate the CVSS score after a number of mitigation actions were applied:

$$CVSS = 10 - mean\left(CVSS_{all\ detected\ vulnerabilities}\right) \quad (1)$$

2.4.2 Return On Response Investment Score

Return on Response Investment (RORI) is a tool used to calculate (a self-named) an index associated to the mitigation actions composing a response plan. The RORI index is used to evaluate optimal plans (by ranking them as a trade-off between their efficiency in stopping potential attacks, and their ability to preserve, at the same time, the best service to legitimate users. Similar metrics such as the "Return of Investment" or the "Return on Security Investment" appear, however, RORI is a variant that has specifically developed for intrusions in Internet Technology (IT) systems. In this work, a modified formula suggested in (Granadillo et al., 2012) is used to calculate it using the following qualities:

- Annual Loss Expectancy (ALE) refers to the financial cost expected from one or more threats, in the absence of applying a mitigation strategy.
- RM is the Risk Mitigation which estimates the effectiveness and coverage of one or more actions in mitigating a threat.

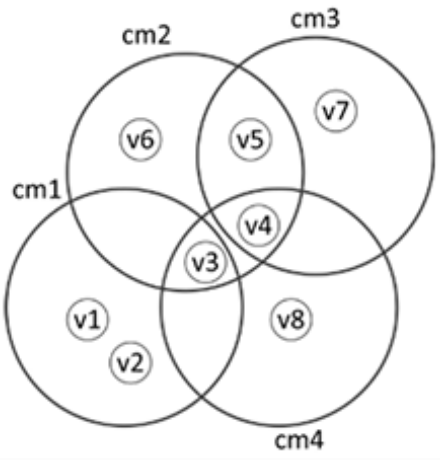


Fig. 2: An example for the calculation of Vulnerability Surface Coverage using four mitigation actions (cm_1, ..., cm_4) that cover eight vulnerabilities (v_1 ,... ,v_8)

- Annual Response Cost (ARC) expresses the expected cost of applying a mitigation strategy and
- Annual infrastructure value (AIC) is a fixed cost associated to the system infrastructure (e.g., cost of equipment, services, etc.), regardless of applying any mitigations. Then,

$$RORI\ Score = \frac{ALE * RM - ARC}{ARC + AIC} \quad (2)$$

2.4.3. Vulnerability Surface Coverage

The Vulnerability Surface Coverage (VSC) of a countermeasure cm_i is the number of vulnerabilities it

covers so VSC ∈ [0,1] (Shameli-Sendi et al., 2018). It can be found in the literature with other names such as Attack Surface Coverage. Disjoint VSC includes the vulnerabilities covered by a single countermeasure, whereas joint VSC refers to the vulnerabilities covered by multiple countermeasures. For example, as shown in Figure 2, disjoint cm_1 has VSC = $\frac{3}{8}$ i.e. v_1,v_2,v_3 out of the 8 total vulnerabilities, while cm_1 and cm_2 have VSC = $\frac{6}{8}$.

2.4.4. Mitigation Deployment Cost

This KPI evaluates the deployment costs of the mitigation actions by considering deployment time, consumed resources and the importance of the device that is affected by the countermeasure as assessed by the network security operator (Li et al., 2018). To calculate it, three quantities are required. First, Deployment Time (DT) which is measured in milliseconds. This is the time required for a mitigation action to be deployed. It can be assessed using historical data and be dynamically updated. Then, the Device Importance (DI), which is arbitrarily assessed by the network security operator considering the specifics of each use case. A value is assigned to each device, where Device Importance ∈ [0,1]. This value is directly mapped to the Asset impact value proposed in the ETSI-TS 102 165 standard (V5.2.3, 2017) as shown in Table 3. Lastly, Resource Consumption (RC) can be imputed in two different ways: If actual resource overhead consumption is available e.g., CPU or RAM or Throughput consumption, the following formula can be used:

$$RC = \frac{resources\,consumed\,by\,device}{total\,amount\,of\,resources} \quad (3)$$

Else, an arbitrarily chosen ranking scheme can be used based on the network operators' expertise i.e RC = *{Very Low:1, Low: 2, Medium: 3, High: 4, Very High: 5}*. Finally, the KPI value is calculated by the following formula:

Table 3
Mapping of the Asset Impact proposed by ETSI to Device Importance

Asset Impact	*Explanation*	*Value*	*Device Importance Value*
Low	The concerned party is not harmed very strongly; the possible damage is low	1	0
Medium	The threat addresses the interests of providers or subscribers and cannot be neglected.	2	0.5
High	A basis of business is threatened and severe damage might occur in this context.	3	1

$$Deployment\ Cost = DT * DI * RC \quad (4)$$

2.5. Application of the mitigation actions to the IoT network

The Mitigation Engine proposed in this work, was designed and built to interact with an SDN controller based on the Open Network Operating System (ONOS) (Berde et al., 2014), described in detail in (Gelenbe et al., 2020). The controller of the SDN system uses the OpenFlow protocol rules (Foundation) to apply the actions outputted by the Mitigation Engine to the network. OpenFlow protocol is supported from the majority of the available SDN controllers (Salman et al., 2016), thus the proposed system can be easily be used with controllers other than ONOS. The actions that are available to the network operator are:

- **Block**: This mitigation action blocks all traffic flows originating from all ports of a specific IP address for a time predefined by the system operator.
- **Block port**: This mitigation action blocks all traffic flows originating from a specific port of a specific IP address for a time predefined by the system operator.
- **Black-list**: This mitigation action blocks all traffic flows originating from all ports of a specific IP address until revoked by the system operator.
- **Redirect to Honeypot** - This mitigation action is designed to reroute traffic between a suspicious host and another network component to a Honeypot in that way that the suspicious host thinks it still connected to the network component originally targeted.
- **White-list**: This mitigation action allows the system operator to undo the effects of a Black-list mitigation action.

3. Proposed Solution method for Multi-objective Knapsack Problems using Pointer Networks and the Normal Normalized Constraint Method

This section starts with a high-level description of the input and output of the proposed module. Then the methodology used to create the proposed Unsupervised IoT-ready Mitigation Engine is showcased. Finally, the description and methodology of the KPI that were used to produce the experimental results are presented.

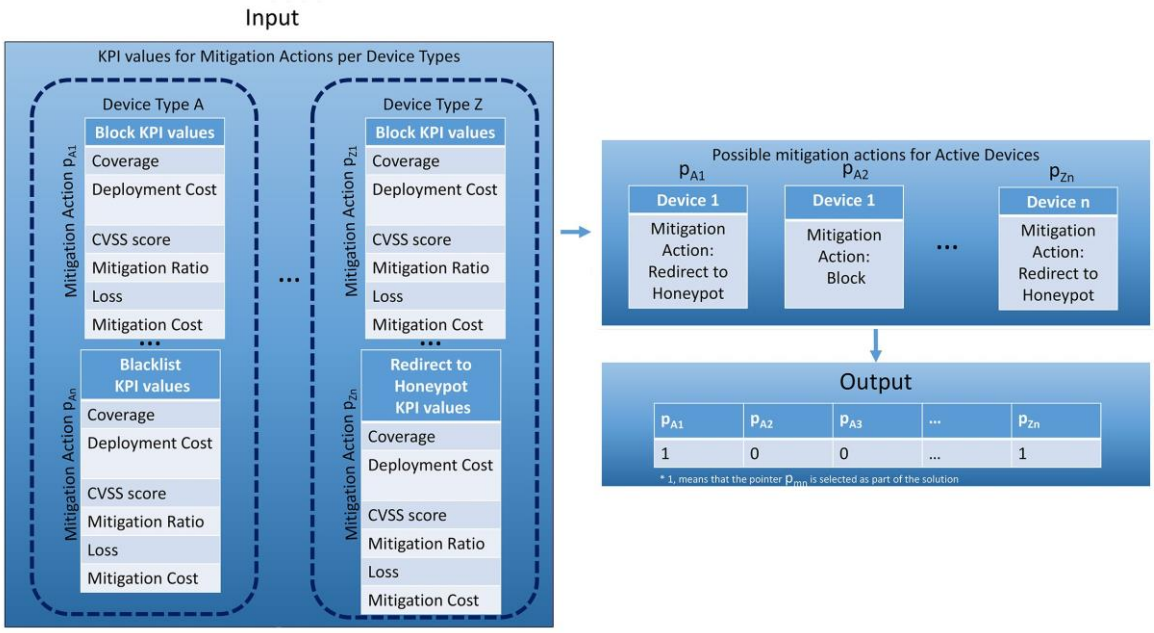


Fig. 3: Input and Output of the proposed the Pointer network mechanism.

3.1. I/O data for the Pointer network mechanism

The following subsection presents the input and output data of the proposed implementation of Pointer networks. For each device type expected to be in the IoT network, a number of different mitigation actions exist described as set of rules. Each of these rules is predefined by the network operator based on past or known attacks and contains a) a unique Device Identifier, b) the Device Type, c) a vulnerability, d) a mitigation action appropriate to this vulnerability and e) values for the variables required to calculate the KPIs that will be used for optimization. A device can have multiple rules, describing different actions to counter an attack against a vulnerability. We assume a mechanism as the one described in (Foremski et al., 2020) is used to monitor device types and identities, while unidentified devices are automatically assigned to an 'unprofiled' category.

It is assumed one or more attacks against *N* devices connected in an IoT network are detected. Let the rule table related to these attacks have *M* rows. The input of the proposed method is an ordered matrix of size *M*x*N* multiple rules for multiple devices: each row is mapped one to one to the original table and contains *N* elements describing the device ID and the values of the variables required to calculate the KPIs that are used for optimization.

Let $\mathcal{M} = \{p_1, p_2, \ldots, p_{|\mathcal{M}|}\}$, be the ordered vector corresponding to each row of the input matrix. Then, the output of the Pointer Networks is a vector $\mathcal{P}$ of size equal to $\mathcal{M}$, $\mathcal{P} = \{c_1, c_2, \ldots, c_{|\mathcal{M}|}\}$, where

$$c_m = \begin{cases} 1, \text{if pointer } p_m \text{ is selected as part of the solution,} \\ 0, \text{if not.} \end{cases}$$

The proposed method takes into account two constrains a) one mandatory that enforces that only a single mitigation action should be applied to each device and b) possible constraints imposed by the operator concerning the values of the KPIs used.

The input and output of the proposed method is shown schematically in figure 3.

3.2. Methodology used for the Unsupervised IoT-ready Mitigation Engine

In the proposed method, the normalized normal constrained (NNC) method (Messac et al., 2003) is used as a general framework to generate the Pareto frontier while each sub-problem inside the NCC process is solved using a deep neural network architecture called Pointer Networks (Vinyals et al., 2015). First, the constrained multi-objective optimization problem is defined: Let x be n items we must choose from, $x_i = 1$ if an item is chosen else $x_i = 0$, and: let μ_l be the lth KPI to be optimized. Then, the following minimization is the target:

$$\min_x (\mu_1(x) \ldots \mu_l(x)), l \geq 2 \quad (5)$$

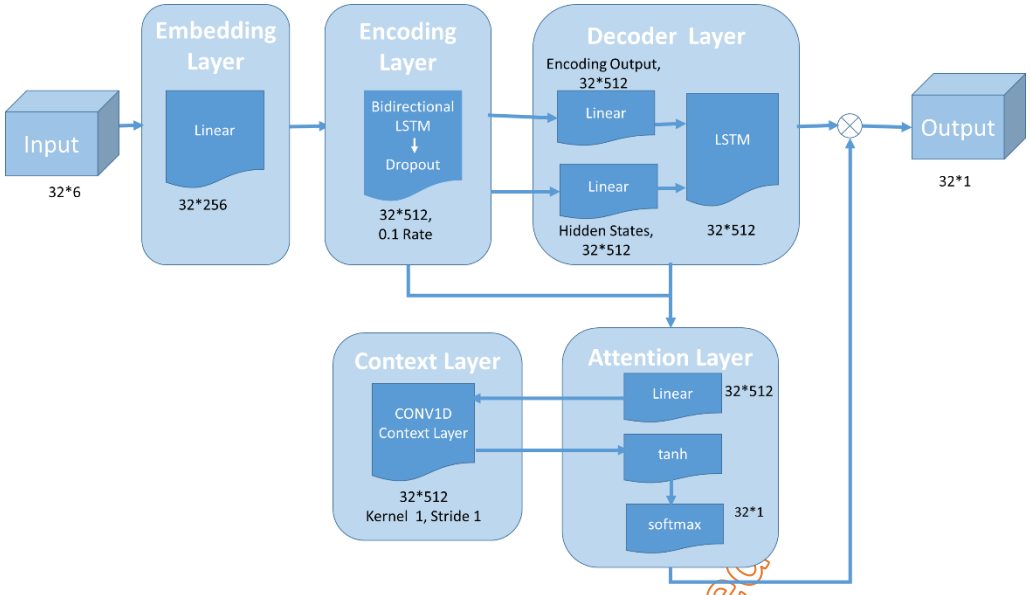


Fig. 4: Architecture of the Pointer Deep Neural Network.

Let $g_j(x)$ and $h_k(x)$, be the r and s inequality and equality constrains of the problem:

$$g_j(x) \leq 0, (1 \leq j \leq r) \quad (6)$$

$$h_k(x) = 0, (1 \leq k \leq s) \quad (7)$$

$$x_i \in 0,1 \quad (8)$$

Equations (6), (7), (8) are inequality, equality and side constraints of the problem respectively. To proceed, n subproblems are be defined and solved, one for each objective function:

$$\min_x \mu_1(x), (1 \geq i \geq l) \quad (9)$$

Each sub-problem is subject to equations (6), (7), (8).

3.2.1. Solution of minimization sub-problems

We propose the use of Pointer Networks to solve equation (9) : Let $P= \{p_1, \dots, p_n\}$ be a sequence of n vectors transformed via a Linear embedding to be used as input, corresponding to $x = \{x_1,\dots,x_n\}$ and $Y = \{y_1,\dots,y_m\}$ be the output sequence associated to P. Pointer networks are based on the Sequence to Sequence Model (Sutskever et al., 2014) with a modified attention mechanism.

Bi-directional Long Short-Term Memory (LSTM) DNN are used to encode and decode the data: Let $E= \{e_1, \dots, e_n\}$ and $D= \{d_1, \dots, d_n\}$ be encoder and the decoder hidden states of the LSTMs. Let f, g be the transformation functions made by the LSTM layers, c be a context vector resulting by an attention mechanism $q(e_1,\dots,e_j)$. Then, the conditional probability calculation can be written as ($i,j \in (1,\dots,n)$):

$$p(y_i \mid y_1,\dots,y_{i-1},P) = g(y_{i-1},d_i,c) \quad (10)$$

$$d_i = h(d_{i-1},y_{i-1},c_i) \quad (11)$$

$$c = q(e_1,..,e_j) \quad (12)$$

$$e_j = f(P_j,e_{j-1}) \quad (13)$$

In Pointer networks, the Encoder and Decoder Layer are connected by the attention mechanism. The context vector c are calculated by the encoder hidden states and the attention weights values a_{ji}, $i,j \in (1,\dots,n)$:

$$c_i = \sum_{j=1}^{n} a_j^i e_j, \quad (14)$$

where

$$a_i^j = softmax(u_i^j) = \frac{exp(u_j^i)}{\sum_{k=1}^{n} exp(u_k^i)} \quad (15)$$

and

$$u_i^j = v^T tanh(W_1 e_j + W_2 d_i) \quad (16)$$

with W_1, W_2 and v parameters that are learned by the network. Finally,

$$p((y_i \mid y_1, ..., y_{i-1}, P; \theta)) = softmax(u^i) \quad (17)$$

provides the conditional probability to choose the pointer y_i at a given iteration of the algorithm.

The SoftMax normalizes the Attention Layer vector values u_i^j to be an output distribution over the dictionary of inputs. After a pointer is chosen, the constrains set by equations (6), (7), (8) are checked and if a violation is found x_i and $p((y_i \mid y_1, ..., y_{i-1}, P; \theta))$ are assigned to zero, else x_i = 1. Figure 4 shows the implementation of the Pointer Deep Neural Network.

3.2.2. Creation of the Pareto Front

After solving the sub-problems defined in equation 9, we can define the following variables: Let $x^{l*} \in 0,1$ be the optimal decision vector and $\mu_l^* = \mu_l(x^{l*}), \mu_l^* \in R^n$ be the lth generic optimal objective. An Utopia Point (Messac et al., 2003) is defined as $\mu^u = [\mu_1^*, ..., \mu_n^*]^T, \mu^u \in R^n$,while the solution space points μ^{l*}, each corresponding to an optimal objective value μ_l^*, are called Anchor Points. The hyper-plane P^u, defined by the anchor points μ_l^* is called the Utopia Plane. Finally, the opposite of the Utopia point, i.e. the point that comprises by the maximum values of the objective function is called the Nadir Point μ^N, where $\mu^N = [\mu_1^N, ..., \mu_l^N]^T$ and each point is found by $\mu^N = max[\mu_1(x^{1*}), ..., \mu_l(x^{n*})]^T$.

Using these quantities, the objective function can be normalized, using:

$$\bar{\mu}_i = \frac{\mu_i - \mu_i(x^{i*})}{s_i}, \quad (18)$$

where

$$m_i \in S, S = \begin{Bmatrix} s_1 \\ . \\ . \\ . \\ s_l \end{Bmatrix} = \mu^N - \mu^u. \quad (19)$$

Let $\bar{N}_k$ be the direction from $\bar{\mu}^{k*}$ to $\bar{\mu}^{n*}$ for $k \in \{1, 2, ..., l-1\}$, as $\bar{N}_k = \bar{\mu}^{n*} - \bar{\mu}^{k*}$. For a predefined number of solutions m_k, a normalized increment of δ_k, is computed along the direction $\bar{N}_k$,

$$\delta_k = \frac{1}{m_k - 1}, (1 \le k \le n-1) \quad (20)$$

The number of the solutions m_k, affects the distribution of the solutions in the Utopia plane. To ensure an even distribution, m_1 is specified and then m_k is calculated by

$$m_k = \frac{m_1 \| \bar{N}_k \|}{\| \bar{N}_1 \|} \quad (21)$$

The set of evenly distributed points in the Utopia plane is evaluated by the following:

$$\bar{X}_{pj} = \sum_{k=1}^{l} a_{kj} \hat{\mu}^k, \quad (22)$$

where

$$0 \le a_{kj} \le 1, \quad (23)$$

$$\sum_{k=1}^{l} a_{kj} = 1. \quad (24)$$

Finally, for each $\bar{X}_{pj}$, a corresponding Pareto solution is obtained by solving the following problem:

$$\min_x \overline{\mu_l}, \quad (25)$$

subject to the starting constraint equations 6, 7, 8 and additionally,

$$\overline{N_k}(\bar{\mu} - \overline{X_{pj}})^T \le 0, (1 \le k \le l-1) \quad (26)$$

$$\bar{\mu} = \{\overline{\mu_1}, ..., \overline{\mu_l}\}, \quad (27)$$

This is solved by using Pointer networks as shown in equations (10) to (17). The values for the objective functions can be calculated using

$$\mu_i = \overline{\mu_i} s_i + \mu_i(x^{i*}), i = 1, 2, .., l. \quad (28)$$

It should be noted, that there are cases while the NCC produces solutions that cover the Pareto front, it can also produce non-optimal solutions especially in highly concave solution regions. The authors of the paper originally proposing NCC (Messac et al., 2003) and propose the use of a filtering mechanism to retain only the Pareto optimal points. Other mechanisms such as the Non-dominated

Sorting algorithm used in NSGA-II Deb et al. (2002) can also be used for the same purpose.

3.2.3. Selection of a single solution from the Pareto Front

To choose a single solution, if required, a modification of the Simple Additive Weighting / Weighted Sum method is proposed: Let $V=\{v_1,...,v_v\}$ all the objective function values for the solutions found, e.g. for v_v the vector is $v_v=\{\mu_{v1},...,\mu_{vl}\}$ and w be a vector containing weights, arbitrarily chosen: $w=\{w_1,...,w_i\}, i=1,2,..,l$ where

$$\sum_{i=1}^{l} w_i = 1 \qquad (29)$$

Then, a solution can be chosen by

$$\underset{v_v \in V}{\operatorname{argmin}} \sum_{i=1,\, j=1}^{l,v} w_i \frac{\mu_{v,l} - min(\mu_{vl})}{max(\mu_{vl}) - min(\mu_{vl})} \qquad (30)$$

In Shameli-Sendi et al. (2018), it is shown that this method outperforms the Weighted Product and the TOPSIS method which are used for the same purpose.

4. Experimental Results

The following section presents the results of the experiments performed to validate the proposed schemes for the Mitigation Engine module. Initially, the steps to prepare the experiments are presented in terms of scenarios and underlying assumptions. Then the parameters used in the various algorithms are presented and finally the experimental results.

4.1. Experimental setup and Algorithm Preparation

A synthetic scenario is used to validate the mitigation engine, in which it is assumed that an IoT network is threatened by multiple attacks and some of the network components are already affected by malicious software. Additionally, it is assumed that one or more Intrusion Detection System components have discovered multiple anomalies in the traffic in various devices in the IoT network: SDN controllers and switches, IoT devices and a mini-PC. More specifically the device types we assume are present in the IoT network are: Routing controllers using the Open Network Operating System (ONOS), Virtual switches using the Open vSwitch, IP cameras, temperature control sensors and mini-PCs operating with Windows 10. The vulnerabilities assumed to be used for the attacks for the experimental scenarios, presented in detail in the National Vulnerability Database (Booth et al., 2013), are shown in table 14 available in Appendix Section B.

To verify the run time of the various algorithms, another series of synthetic data sets were used. Details on the creation data sets are available in Section 4.6. This was to ensure that the various experiment runs where independent in terms of data used: N devices with 4 mitigation rules each were created using random generators with specific range. Specific parameters are available in Table C in the appendix.

For the purposes of evaluations, results for three algorithms, the Non-Dominated Sorting Genetic algorithm II (NSGA-II) (Deb et al., 2002), the Multi-objective Evolutionary Algorithm Based on Decomposition (Zhang and Li, 2007) and the Improved Harmony Search (IHS) (Mahdavi et al., 2007) are generated.

All three are AI algorithms that belong to the family of Evolutionary algorithms. Based on the literature review available in Section 2, NSGA-II is the SoA for mitigation action selection using MO. The three algorithms respond to the three major approaches in evolutionary algorithms (Liu et al., 2020): Pareto dominance-based (NSGA-II), scalarizing function-based (MOEA/D), and indicator-based algorithms (IHS). Details on the parameters used to train all algorithms are available in Appendix C.

4.2. Evaluation Results

To evaluate the effectiveness of the proposed approach, several experiments were performed with data-sets of different sizes. To summarize, the results show that in most cases, the proposed method produces equal or better solutions than the SoA methods, with less time required by the Pointer Networks in most cases. All models were created using the parameters explained in 4.1.

For each experiment, the modified Weighted Sum method procedure described in Section 3.2 is used to choose a single best solution. Apart from the KPI results for each case, values for the Hypervolume Indicator are presented: It is a set measure widely used to measure the diversity of the solutions offered by algorithms in Multi-Objective problems and their closeness to the Pareto Front (Liu et al., 2020): An algorithm with a higher Hypervolume Indicator *H, H∈[0,1],* produces a more diverse solution set and thus offers more choices to the decision-maker. Such a metric is needed since MO problems do not have a single solution and the notion of a correct solution is not applicable, so metrics such as accuracy can not be applied. Hypervolume which is also known as the S metric is the most widely employed such metric in the literature (Riquelme, 2015).

4.3. Experiment A: Small Sized Data Set, 10 Devices with 40 rules

In this experiment 10 devices were assumed to be under attack: 2 of each type i.e ONOS SDN, Virtual switches

using Open vSwitch, IP cameras, Temperature control sensors and mini-PCs. As already mentioned in the introduction of this section, each experiment was repeated 100 times. The results of a single experiment can be seen in Figure 5.

On average the Pointer network approaches produces solutions with a lower Deployment Cost and higher ROSI score compares to the other methods, while always covering all vulnerabilities. On average the MOEA/D method produces solutions with lower CVSS scores,

Table 4

Mean and Standard Deviation for the four KPIs used for optimization, number of mitigation actions for each Algorithm (n = 100 experiment iterations). The min/max next to each metric indicates the target.

	Algorithm \ Score ($\mu \pm \sigma$)	Deployment Cost (min)	Vulnerability Coverage (max)	CVSS Score (min)	ROSI Score (max)	Mitigation Actions (max)
Experiment A	IHS	32.647 ± 8.168	91.782 ± 8.137	3.481 ± 0.709	383.678 ± 47.301	9.178 ± 0.814
	MOEA/D	33.048 ± 7.602	85.631 ± 10.274	3.447 ± 0.762	368.471 ± 48.744	8.563 ± 1.027
	NSGA-II	32.598 ± 7.656	91.085 ± 8.112	3.522 ± 0.676	378.512 ± 47.118	9.108 ± 0.811
	Proposed Approach (PointerNet)	42.886 ± 10.497	95.256 ± 12.463	3.413 ± 0.704	411.197 ± 52.727	9.626 ± 1.246
Experiment B	IHS	32.968 ± 1.659	93.915 ± 1.439	3.389 ± 0.119	10081.534 ± 170.102	234.787 ± 3.598
	MOEA/D	32.854 ± 1.542	72.398 ± 2.462	3.365 ± 0.137	7796.169 ± 294.415	180.996 ± 6.156
	NSGA-II	33.108 ± 1.463	87.613 ± 1.884	3.369 ± 0.133	9434.483 ± 208.497	219.032 ± 4.711
	Proposed Approach (PointerNet)	40.948 ± 7.498	96.217 ± 11.139	3.386 ± 0.288	10386.885 ± 1129.744	240.541 ± 27.848
Experiment C	IHS	32.888 ± 1.154	93.627 ± 0.81	3.371 ± 0.087	20066.122 ± 204.61	468.135 ± 4.049
	MOEA/D	32.265 ± 1.171	71.219 ± 1.464	3.32 ± 0.114	15353.518 ± 329.588	356.094 ± 7.32
	NSGA-II	32.679 ± 1.124	82.582 ± 1.472	3.333 ± 0.1	17781.02 ± 349.256	412.909 ± 7.362
	Proposed Approach (PointerNet)	40.748 ± 7.607	96.221±10.879	3.347 ± 0.263	20743.963 ± 2233.278	481.104 ± 54.394

The Pointer Network Solution and the NSGA - II solution sets both contain the solution with the highest Coverage and smallest Deployment Cost combination while the highest ROSI SCORE and smallest CVSS score is contained in the Pointer Network and the MOEA/D solutions set. The evolutionary algorithms tend to produce sets that do not provide mitigation actions for all attacks more often than the Pointer networks.

Pointer networks on average, produce solution sets with a larger value of Vulnerability Coverage and ROSI score, smaller values for the CVSS score and mitigation actions for more devices compared to the Evolutionary Algorithms as shown in Table 4 .

Additionally, the Pointer Network results have a larger Hypervolume Indicator Value compared to other algorithms as seen in Table 5.

Finally, the modified Weighted Sum method procedure is used to choose a single best solution from each experiment. The aggregated results are shown in Table 6: however, there is 0.79% percentage difference with the average score of the solutions produced by the Pointer Networks.

Table 5

Mean and Standard Deviation for the Hypervolume Indicator for each Algorithm (n = 100 experiment iterations), for experiment 1.

Algorithm \ Score ($\mu \pm \sigma$)	Hypervolume Indicator Value
IHS	0.3784± 0.196
MOEA/D	0.309± 0.179
NSGA-II	0.383± 0.166
Proposed Approach (PointerNet)	0.565± 0.166

Table 6

Mean and Standard Deviation for the four KPIs used for optimization, number of mitigation actions for each Algorithm (n = 100 experiment iterations), for the final solution chosen in each experiment.

	Algorithm \ Score ($\mu \pm \sigma$)	Deployment Cost (min)	Vulnerability Coverage (max)	CVSS Score (min)	ROSI Score (max)	Mitigation Actions (max)
Experiment A	IHS	28.315 ± 7.974	95.455 ± 7.038	3.457 ± 0.699	403.788± 41.192	9.545 ± 0.704
	MOEA/D	30.144 ± 7.263	91.919 ± 8.533	3.401 ± 0.743	396.465 ± 38.274	9.192 ± 0.853
	NSGA-II	30.495 ± 7.177	96.364 ± 5.042	3.458 ± 0.665	409.354 ± 29.109	9.636 ± 0.504
	Proposed Approach (PointerNet)	23.949 ± 2.766	100 ± 0	3.428 ± 0.659	430	10
Experiment B	IHS	31.314 ± 1.824	94.56 ± 1.425	3.385 ± 0.119	10150.2 ± 168.303	236.4 ± 3.562
	MOEA/D	32.229 ± 1.45	72.924 ± 2.362	3.362 ± 0.14	7854.49 ± 283.765	182.31 ± 5.906
	NSGA-II	32.812 ± 1.42	88.768± 1.814	3.353 ± 0.127	9556.48 ± 205.139	221.92 ± 4.534
	Proposed Approach (PointerNet)	26.179 ± 1.991	99.932 ± 0.68	3.378± 0.108	10746.77± 32.3	249.83± 1.7
Experiment C	IHS	31.716 ± 1.289	94.074 ± 0.971	3.367 ± 0.08	20167.407 ± 234.586	470.37 ± 4.853
	MOEA/D	31.854 ± 1.149	71.415 ± 1.517	3.317 ± 0.118	15391.074 ± 339.151	357.074 ± 7.585
	NSGA-II	32.414 ± 1.195	83.481 ± 1.48	3.315 ± 0.094	17991.481 ± 344.583	417.407 ± 7.402
	Proposed Approach (PointerNet)	25.983 ± 1.846	100 ± 0	3.348 ± 0.089	21500 ± 121.34	500 ± 0

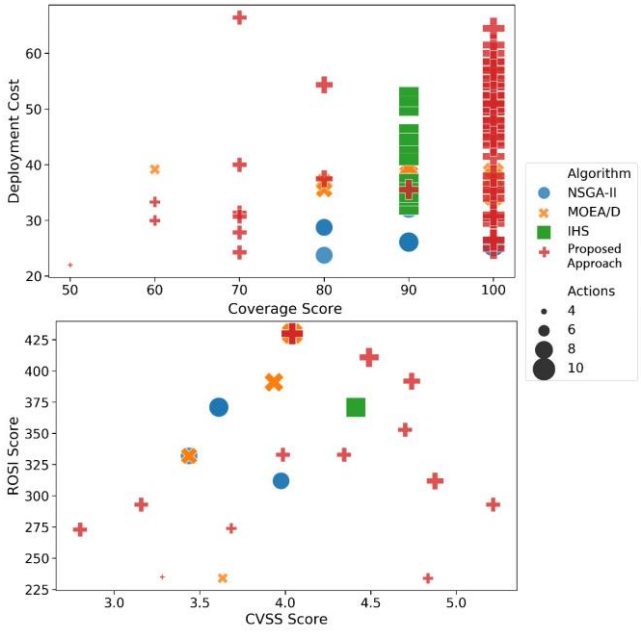


Fig. 5: Results of an instance of experiment 1 for a) Coverage Score (target is to maximize) vs Deployment Cost (target is to minimize) (up) and b) CVSS Score (target is to minimize) vs ROSI score (target is to maximize). It should be noted that figures include the entire solution set which contains both optimal and suboptimal solution if viewed from the prospective of the operator.

4.4. Experiment 2: Medium Sized Data Set, 250 Devices with 1000 rules

In this experiment, 250 devices were assumed to be under attack, i.e., 50 of each type where used. Figure 6shows the results of a single experiment: The Pointer Network solution set contains the solution with the highest Coverage /smallest Deployment Cost combination and the solution with the highest ROSI SCORE /smallest CVSS score.

The Evolutionary Algorithm solutions contain almost no variation in terms of the Vulnerability Coverage and the CVSS score. Pointer networks on average, produce solution sets with a larger value of Vulnerability Coverage and ROSI score, and mitigation actions for more devices compared to the Evolutionary Algorithms as shown in Table 4, while MOEA/D results have the lowest average CVSS score and Deployment Cost.

Additionally, the Pointer Network results have a larger Hypervolume Indicator value compared to HIS, NSGA-II and MOEA/D algorithms as seen Table 7.

Finally, the aggregated results of choosing a single 'best' solution via a weighted sum method are shown in

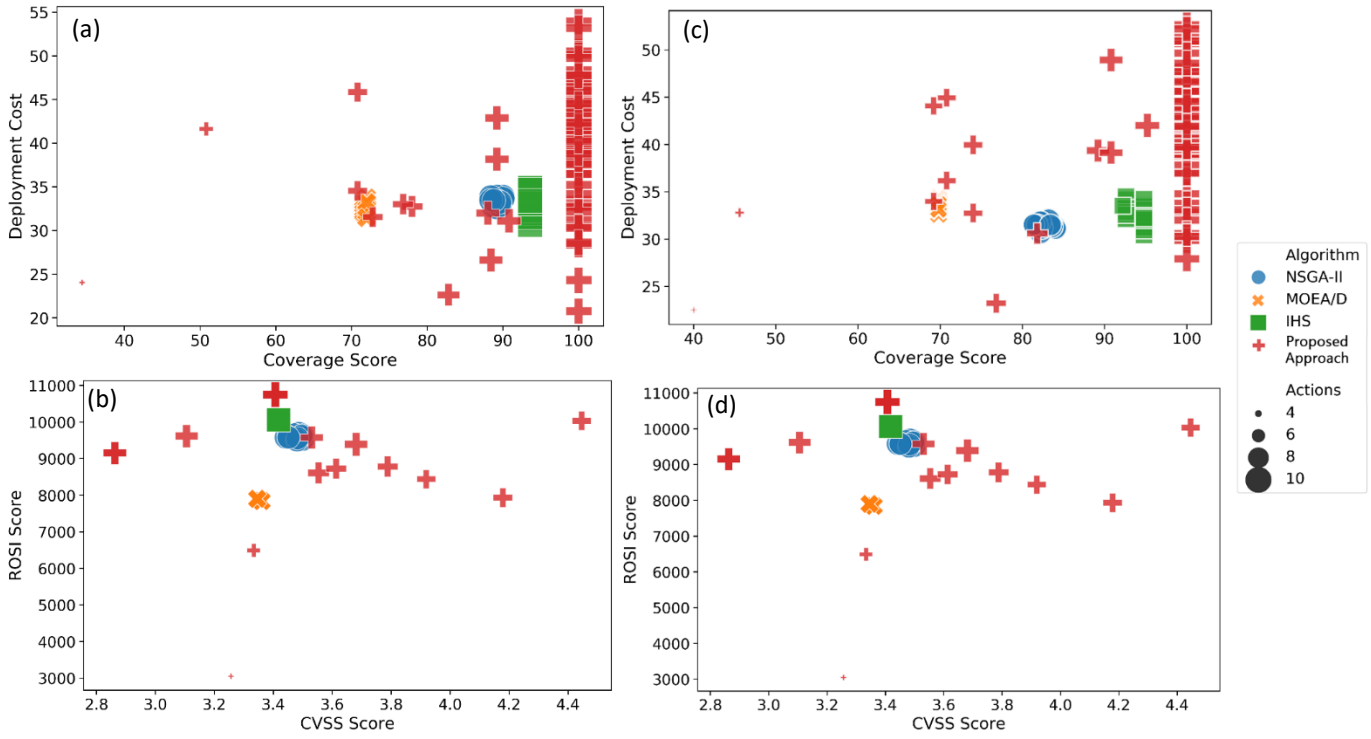


Fig. 6: Results of an instance of experiment 2 for (a) Coverage Score (target is to maximize) vs Deployment Cost (target is to minimize) (up) and b) CVSS Score (target is to minimize) vs ROSI score (target is to maximize). Results of an instance of experiment 3 for c) Coverage Score (target is to maximize) vs Deployment Cost (target is to minimize) and d) CVSS Score (target is to minimize) vs ROSI score (target is to maximize). It should be noted that figures include the entire solution set which contains both optimal and suboptimal solution if viewed from the prospective of the operator.

Table 6: Pointer networks on average, produce solution sets with a larger value of Vulnerability Coverage, lowest Deployment Cost, Highest ROSI score and mitigation actions for more devices compared to the Evolutionary Algorithms. NSGA-II results have the lowest average CVSS score. The percentage difference of the Pointer Network solutions compared to the best solutions is 0.74% for the CVSS score.

4.5. Experiment 3: Large Size Data Set, 500 Devices with 2000 rules

In this experiment, 500 devices were assumed to be under attack, i.e., 100 of each type where used. The results are very similar to those presented in section 4.4: Figure 6 shows the results of a single experiment. The Pointer Network solution sets contains the solution with the highest coverage /smallest deployment cost combination and the solution with the highest ROSI SCORE /smallest CVSS score.

The Evolutionary Algorithm solutions again contain almost no variation in terms of the Vulnerability Coverage and the CVSS score. Pointer networks on average, produce solution sets with a larger value of Vulnerability Coverage and ROSI score, and mitigation actions for more devices compared to the Evolutionary Algorithms as shown in Table 4, while MOEA/D results have the lowest average CVSS score and Deployment Cost.

Additionally, the Pointer Network results have a larger Hypervolume Indicator value compared to HIS, NSGA-II and MOEA/D algorithms as shown in Table 8.

Finally, the aggregated results of choosing a single 'best' solution via a weighted sum method are shown in table 6: Pointer networks on average, produce solution sets with a larger value of Vulnerability Coverage, lowest Deployment Cost, Highest ROSI score and mitigation actions for more devices compared to the Evolutionary Algorithms. NSGA-II solutions have the lowest average CVSS score. The percentage difference of the Pointer Network solutions compared to the best solutions is 0.99% for the CVSS score.

Table 7
Mean and Standard Deviation for the Hypervolume Indicator for each Algorithm (n = 100 experiment iterations), for experiment 2.

Algorithm \ Score ($\mu \pm \sigma$)	Hypervolume Indicator Value
IHS	0.049 ± 0.0358
MOEA/D	0.02 ± 0.0149
NSGA-II	0.039 ± 0.028
Proposed Approach (PointerNet)	0.114 ± 0.045

4.6. Run-Time comparison results

In the results presented in sections 4.3,4.4,4.5, the datasets contained multiple devices of each type. In this section, the results present the run-time of the algorithms using data-sets with unique devices and gradually growing size of [40,100,500,1000,2000] rules in total, are presented to ensure variations in the experiments. All experiments were repeated for *n = 100* times and the aggregated mean runtime is presented.

All the algorithms presented in the following section were implemented using the Python Language. Pointer Networks were implemented using the PyTorch Framework (Paszke et al., 2019), while the Evolutionary

Algorithms were implemented using the Pygmo Framework (Biscani and Izzo, 2020). All experiments were performed using a desktop computer running Windows 10 OS, with an Intel i5-7600 processor @ 3.5 GHz and 8 GB of RAM.

The results for the run time required for each algorithm are shown in figure 8: For smaller data-sets, the Pointer Network algorithm requires less time to finish compared to all three Evolutionary Algorithms. However, as the data set size grows the run-time of the Pointer Network seems to approach that of the MOEA/D, which shows better Run Times compared to IHS and NSGA-II. Table 9 contains comparative results between the Pointer Network results and the algorithm with the second-best performance i.e., MOEA/D. Apart from the mean run time and the accompanying s.d value the Percentage difference is calculated to help quantify the difference (show in Table 10). Percentage difference is calculated by the following formula.

$$\frac{|x_1 - x_2|}{0.5*(x_1 + x_2)}$$

Table 9
Comparison of mean and standard deviation of Run Time value for the Pointer Network and the algorithm with the second-best results (MOEA/D) ($\mu \pm \sigma$).

Rules	Pointer Network mean Run Time in seconds	MOEA/D mean Run time in seconds	Percentage Difference
40	5.7896 ± 0.072543	27.3059 ± 0.413316	130%
100	30.6114 ± 7.331567	44.8527± 0.728498	37.74%
500	148.8513 ± 20.168516	158.1334 ± 1.445462	6.047
1000	292.7027 ± 32.185371	313.7667 ± 2.847596	6.94
2000	592.9054 ± 50.300889	622.5335 ± 3.752541	4.87

Table 8
Mean and Standard Deviation for the Hypervolume Indicator for each Algorithm (n = 100 experiment iterations), for experiment 3.

Algorithm \ Score ($\mu \pm \sigma$)	Hypervolume Indicator Value
IHS	0.065043 ± 0.0342
MOEA/D	0.029961 ± 0.0145
NSGA-II	0.047773 ± 0.024911
Proposed Approach (PointerNet)	0.155188 ± 0.05689

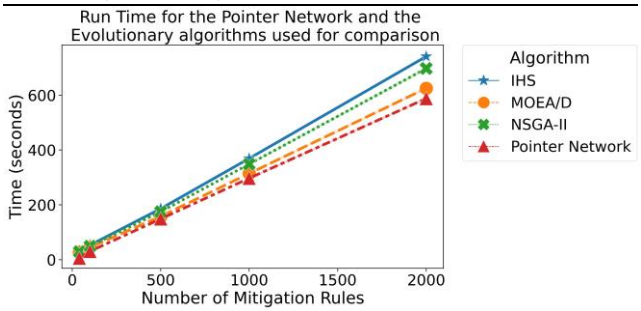


Fig. 8: Mean Run Time of the Pointer Network and the Evolutionary algorithms used for comparison, for experiments with different data set sizes.

5. Conclusions and Further Challenges

This paper presented an automated AI based method that enables the decision of optimal strategies against threats and attacks faced by an IoT network. The timely application of appropriate mitigation actions against the threats faced by IoT networks, based on well-defined KPIs, is a crucial part of any such system. This is enabled by a beyond State-of the-Art AI mechanism that is based on Pointer Deep Neural Networks.

Two serious challenges were faced during the design process of the proposed approach. Initially, to discover an appropriate method to model mitigation actions of the various devices in a manner appropriate to be used as an input to the Pointer Networks. This was overcome by using an approach that maps the mitigation actions to the vulnerabilities of the devices that are expected to be part of the IoT network. The second challenge was to determine a set of KPIs that would successfully describe the various positive and negative impacts of a set of mitigation actions which was overcome by an extensive review of the available literature. The resulting methodology can be easily used to describe the vulnerabilities of any device. Using this methodology, the mitigation mechanism can be expanded to describe more mitigation actions and additional KPIs. The advantage of this approach is that the Deep Learning solution proposed, scales better than other existing AI solutions while producing high-quality diverse solutions. An additional technical challenge was to ascertain a way to properly transform the problem to be solved by PointerNets and ensure that a well populated

Pareto front was produced. This was overcome by employing the NCC method to create the final solution set.

Experimental results validate that the proposed scheme is more efficient in terms of time required to run and in terms of quality of solutions produced in that time compared to three other SoA AI algorithms. The experiments were performed using a data set that contains real, recently discovered vulnerabilities. The results show a consistent behaviour of the proposed scheme in experiments with data sets of varying size. In the most complex scenario examined in our experiments, the results from the Pointer Network are better for KPI concerning the apart from the CVSS score where it under-performs the difference is small (approx. 1 %). Moreover, it is shown by experiments that the Pointer Network approach is at least 4 % faster compared to the SoA methods used for comparison.

The proposed method is strongly based on the normalized normal constrained method which while easy to apply is known to suffer for two major drawbacks: First, the method is heavily dependent on the choice of the Utopia Points and second is that in some cases it can produce non-pareto solutions that then require filtering which adds latency to the system. In order to overcome these limitations, as follow-up work, we aim to test additional methods with Pointer Networks for the creation of the Pareto front such as Decomposition Methods and variations of the NCC. Additionally, we plan to investigate the beam search procedure as proposed in other applications of Pointer Networks, to try and achieve faster convergence rates. Finally, we plan to further test the proposed approach by deploying and using it in real conditions, verifying that it can handle challenges such as large scale IoT networks and systems that require low latency even under poor resource utilization cases.

Table 10
Percentage Difference of the mean KPI results produced by Pointer Networks compared to the second-best results.

KPI(target)	Percentage Difference (%)
Deployment Cost (min)	-19.8721
Vulnerability Coverage (max)	6.01
CVSS (min)	0.99
ROSI (max)	6.39

Appendix

A. Details on data used and performance achieved in the papers presented in the literature review concerning on the selection of mitigation actions for cyber-attacks

Table 11 presents details concerning the data and the experimental results of the papers shown in Table 1 concerning the methods applied for the selection of mitigation actions against cyber-attacks. It should be noted that in many cases the Authors simply present the application of their proposed methods without evaluating it against other methods making it hard to extract meaningful insights concerning its' performance.

B. Details on the vulnerabilities and mitigation rules used for the experiment

Table 12 contains the details of the vulnerabilities and relevant mitigation rules used for the experiments presented in Section 4. These parameters can be used to recreate the datasets used in the experiments presented in this paper.

C. Details on training parameters of the Algorithms used for the experiments

The DNN models were trained for 30 epochs on 100000 examples produced by solving the SO problem using synthetic data. Solutions for the training data were produced by a simple Genetic Algorithm: In (Bello et al., 2017)] 10000 instances were used while the authors of (Gu and Hao, 2018), use 1000 instances of training data. The number of epochs was decided based on experimentation: In all cases, the loss function converges to a stable value between 25 to 30 epochs. A simplex lattice design (Pescador-Rojas and Coello, 2018) with 50 number of points was used to generate the weights required for the NNC method.

The evolutionary models for NSGA-II and MOEA/D were implemented using a solution population of 50, evolving for 150 epochs based on (Hasan et al., 2018). The IHS was implemented using a solution population of 10 and 10000 generations based on the recommendations of (Mahdavi et al., 2007) and (Ouyang et al., 2017). The rest of the parameters used in training the Pointer networks based in (Gu and Hao, 2018) and the parameters used for the evolutionary algorithms are available in table 13.

Table 13 Parameters used to train the Pointer Networks, parameters used for the Evolutionary Algorithms and parameters used for the synthetic data-sets used for the Run Time experiments.

Case	Parameters
Pointer Networks	Batch Size: 512, Epochs: 30, Optimizer: Adam, lr = 0.001, Embedding size: 128, hidden layer size: 256, LSTM -Layers: 2
NSGA -II	Crossover Probability: 0.95, Mutation probability: 0.01, Distribution index for mutation: 10, Distribution index for mutation: 50
MOEA/D	Weight generation: "grid", Decomposition method: "tchebycheff", Neighbours: 20, Crossover parameter in the Differential Evolution operator: 1, Parameter for the Differential Evolution operator: 0.5, Distribution index used by the polynomial mutation: 20, Chance that the neighbourhood is considered at each generation: 0.9
IHS	Probability of choosing from memory: 0.85, Pitch Adjustment: [0.35,0.99], Distance bandwidth: [1e-05, 1]
Run Time Data Set	Device Name: Unique id of length 10, Dev ID: Unique ID of length 5, Vuln ID: Unique ID of length 10, CVSS score: Random float between 2 and 10 with max 2 decimal, Device Importance : Random choice from the set [0.25,0.5,0.75,1], Loss : Random choice from the set [10,15,20,25,30,35,...100], Mitigation Time : Random Float between 1 and 4

Table 11 Details concerning the data and the experimental results of the papers shown in the literature review concerning the selection of mitigation actions to counter cyber-attacks.

Paper	Dataset Used	Results
Gonzalez-Granadillo et al. (2015)	Synthetic dataset of 11 devices describing a Critical Infrastructure Control System (Power Plant) and 6 different user types . 2 Attacks with 13countermeasures were examined.	Proposed method was not compared to other methods.
Granadillo et al. (2012)	Synthetic dataset of 12 devices describing a Mobile Money Transfer Service. Two Attacks with 9 countermeasures were examined.	Proposed method was not compared to other methods.
Gonzalez - Granadillo et al. (2017)	Synthetic dataset of 9 device types describing a Power supply SCADA network is composed of 13,000 energy stations, 6,000 of which are controlled by a central system. Three Vulnerabilities indexed by CVE number along with three countermeasures were used.	Proposed method was not compared to other methods.
Chung et al. (2013)	A network of 7 servers and 3 VMS was created.11 Countermeasures to ten Vulnerabilities indexed by CVE number were used.	Three variants of the proposed method were compared: Results showed that deploying the solution on the network controller was more efficient in terms of detection delay and resource use compared to deploying it on a proxy or a mirror server.
Kotenko and Doynikova (2016)	A network of 4 servers and 6 workstations was created.4 Countermeasures to 15 Vulnerabilities indexed by CVE number were used.	Proposed method was not compared to other methods.
Doynikova and Kotenko (2018)	Experiments on networks sized with 10 to 500 web servers and firewalls were simulated. 13 Countermeasures were used.	Proposed method was compared to a strategy where no countermeasures were selected, resulting to 80% less economical losses.
Wang et al. (2013)	4 device types (VOIP phones, firewalls, servers and workstations) with 10 vulnerabilities indexed by CVE number were simulated. 35 countermeasures were utilized.	Proposed method was not compared to other methods.
Zonouz et al. (2009)	2 device types (servers, work stations) were simulated and scenarios with up to 330k hosts were examined. The number of adversarial actions and responses was modified throughout experimentation.	Proposed method was compared to a static greedy response selection method. Results show that the proposed method that minimize system damage and intrusion response cost.
Zonouz and Haghani (2013)	Smart Grid network with two control planes and 103 Cisco PIX firewall rules. Authors do not provide information concerning specific attacks, device types or countermeasures.	Results show that the proposed method can match performance of human network security operator after training.
Miehling et al. (2015)	Synthetic dataset describing a small network with 4 devices, 12 vulnerabilities and 4 countermeasures.	Proposed method was not compared to other methods.
Poolsappasit et al. (2012)	6 servers, a firewall and 2 workstations with 14 vulnerabilities indexed by CVE number were simulated. 13 countermeasures were utilized.	Authors propose a SO and a MO version of a Genetic Algorithm and show through experimentation that a Multi-objective approach requires less tuning and the relavant problem is solved faster compared to the Single-Objective version.
Dewri et al. (2012)	Authors simulate a small network 3 servers, a firewall and 1 workstation with 8 vulnerabilities indexed by CVE number were simulated. 19 countermeasures are demonstrated.	A Single-Objective Genetic Algorithm (GA) is proposed to select optimal countermeasures. Experiments show that a Multi-objective approach based on the NSGA-II requires less tuning and is more robust under an evolving attack .
(Garzia et al., 2017)	A synthetic dataset of 13 device types is used. No specific vulnerabilities or attacks are examined	Proposed method was not compared to other methods.

(Roy et al., 2012)	Authors used a modelling tool called SHARPE to model the attack - countermeasure tree of a SCADA system and applied their proposed method there. Additionally some experiments where performed on a network with 2 attacker hosts, 7 target hosts and a set of 12 vulnerabilities per host.	The authors show that a technique called Integer programming can compete with a Simple GA in the run-time required to find the optimal solution Set.
(Chehida et al., 2020)	The authors present a list of 11 attacks and 10 countermeasures along with their cost and other security related parameters and then try to find a optimal policy balanced between cost and risk in a finite time duration.	Proposed method was not compared to other methods.
Li et al. (2018)	A Synthetic dataset of 12 devices (Servers, Workstations, Routers, Firewalls). The devices suffer by 13 different vulnerabilities indexed by CVE. No specific countermeasures are examined.	The authors experimentally show that the countermeasure selection is better than a random selection of solutions.
Rachedi and Bensli-mane (2016)	A Synthetic dataset describing a Wireless Sensor Network Data. No attack is assumed. The authors try to solve the problem of balancing various Security parameters and the Quality of Service (QoS) parameters.	Proposed method was not compared to other methods.
Lee et al. (2017)	Authors use a synthetic dataset comprised of 500 security countermeasures, each with an assigned cost and contribution to decrease Risk to the system. No attack is modeled, instead the problem of balancing cost vs Risk decrease is examined.	The proposed method based on the MO NSGAII algorithm is shown to outperform a SO GA in terms that it produces set of solutions that include and dominate the solution produced by the GA.
Hasan et al. (2018)	A Synthetic dataset describing a Energy Delivery system using SDN comprised by 6 different component types. No attack is assumed. The authors try to solve the problem of balancing multiple security and QoS parameters.	Proposed method was not compared to other methods.
Enoch et al. (2019)	A simulation of a network varying size (10 to 300 hosts) was used, comprised by 5 different device types with 12 vulnerabilities indexed by CVE number were simulated. 13 countermeasures were utilized.	Proposed method was not compared to other methods. The authors carried a series of experiments that show that the nature based algorithm used (NSGA-II) required tuning concerning both the number of generations and the population size in order to produce a Pareto solution set.
ShameliSendi and Dagenais (2015)	A network with 11 servers and approx. 150 users is simulated. The Network is under 4 different attacks linked to CVE ids that can be countered by 30 measures.	Proposed method was not compared to other methods.
ShameliSendi et al. (2018)	Demonstration on a network comprised of 9 cloud servers of 3 different types connected by Openstack. Servers suffer by 15 different vulnerabilities indexed by CVE ids that are countered by 59 security measures.	The authors examine three methods to select a single solution for a MO solution set: Simple Additive Weighting, Weighted Product method and the TOPSIS method and show that the first method outperforms the rest.
Viduto et al. (2012)	Authors model a non-specific network that suffers by 10 different vulnerabilities indexed by CVE ids that can be exploited by 5 different attacks and countered by 24 security measures. The authors try to balance two objectives: Risk and Cost.	Experiments show that using a Tabu Search Algorithm can obtain 50% of the Pareto front solution in a fraction of the time required to perform an exhaustive search, however it fails to find the entire Pareto front even after a large number of iterations.

Table 12 Mitigation Actions and relevant information required to calculate the KPIs shown in the experimental results

Device Name	Device Import.	Loss	CVSS ID	Description	CVSS	Mitigation Action	Mit. Re-sources	Mit. Time	Dep. Cost
Open VSwitch	50	40	CVE 2017 9265	Buffer over-read issue that can enable an attacker to cause a denial of service attack (DOS).	7.5	Honeypot	2	0.25	25
						Block	1	0.2	10
						Blacklist	1.5	0.2	15
						Block Port	1.8	0.2	18
			CVE 2018 17205	This allows a flow update action causing an assertion failure, that leads to incorrect flows or DOS.	5	Honeypot	2	0.25	25
						Block	1	0.2	10
						Blacklist	1.5	0.2	15
						Block Port	1.8	0.2	18
IP cam	25	20	CVE 2018 19080	This allows the attacker to inject scripts through a Cross Site Scripting attack.	4.3	Honeypot	2	0.25	12.5
						Block	1	0.2	5
						Blacklist	1.5	0.2	7.5
						Block Port	1.8	0.2	9
			CVE 2018 19081	This allows the attacker to gain total control of the camera including execution of OS commands.	10	Honeypot	2	0.25	12.5
						Block	1	0.2	5
						Blacklist	1.5	0.2	7.5
						Block Port	1.8	0.2	9
			CVE 2018 19082	This allows the attacker to conduct stack-based buffer overflow attacks via the IPv4Address field.	7.5	Honeypot	2	0.25	12.5
						Block	1	0.2	5
						Blacklist	1.5	0.2	7.5
						Block Port	1.8	0.2	9
ONOS	75	40	CVE 2018 1000615	This allows the remote crashing of services offered, via a normal or forged switch connected in the network.	5	Honeypot	2	0.25	37.5
						Block	1	0.2	15
						Blacklist	1.5	0.2	22.5
						Block Port	1.8	0.2	27
			CVE 2018 12691	This allows attackers to bypass network access control via data plane packet injection.	4.3	Honeypot	2	0.25	37.5
						Block	1	0.2	15
						Blacklist	1.5	0.2	22.5
						Block Port	1.8	0.2	27
Windows PC	100	60	CVE 2019 1368	Erroneous Windows security configurations related to debugging can lead to security feature bypass.	2.1	Honeypot	2	0.25	50
						Block	1	0.2	20
						Blacklist	1.5	0.2	30
						Block Port	1.8	0.2	36
			CVE 2019 1359	Vulnerability caused by improper handling of objects in memory allowing remote code execution.	9.3	Honeypot	2	0.25	50
						Block	1	0.2	20
						Blacklist	1.5	0.2	30
						Block Port	1.8	0.2	36
Temp. Controller	25	60	CVE 2017 14020	This allows arbitrary code execution by not sanitizing user input, which is exploited to execute malicious code.	9.3	Honeypot	2	0.25	12.5
						Block	1	0.2	5
						Blacklist	1.5	0.2	7.5
						Block Port	1.8	0.2	9

References

Ahemd, M.M., Shah, M.A., Wahid, A., 2017. Iot security: A layered approach for attacks & defenses. 2017 International Conference on Communication Technologies (ComTech) , 104–110.

Akyildiz, I.F., Lee, A., Wang, P., Luo, M., Chou, W., 2014. A roadmap for traffic engineering in sdn-openflow networks. Computer Networks 71, 1–30. DOI:https://DOI.org/10.1016/j.comnet.2014.06.002.

Al-Janabi, T.A., Al-Raweshidy, H.S., 2018. A centralized routing protocol with a scheduled mobile sink-based ai for large scale i-iot. IEEE Sensors Journal 18, 10248–10261. DOI:10.1109/JSEN.2018.2873681.

Alahakoon, D., Yu, X., 2016. Smart electricity meter data intelligence for future energy systems: A survey. IEEE Transactions on Industrial Informatics 12, 425–436. DOI:10.1109/TII.2015.2414355.

Ali, M.H., Jaber, M.M., Abd, S.K., Rehman, A., Awan, M.J., Damaševičius, R., Bahaj, S.A., 2022. Threat analysis and distributed denial of service (ddos) attack recognition in the internet of things (iot). Electronics 11, 494. DOI:10.3390/electronics11030494.

Awan, M.J., Masood, O.A., Mohammed, M.A., Yasin, A., Zain, A.M., Damaševičius, R., Abdulkareem, K.H., 2021. Image-based malware classification using vgg19 network and spatial convolutional attention. Electronics 10.

Back, T., Hammel, U., Schwefel, H.P., 1997. Evolutionary computation: Comments on the history and current state. IEEE transactions on Evolutionary Computation 1, 3–17.

Bello, I., Pham, H., Le, Q.V., Norouzi, M., Bengio, S., 2017. Neural combinatorial optimization with reinforcement learning, in: International Conference on Learning Representations, ICLR 2017, pp. 1–5.

Bera, S., Misra, S., Vasilakos, A.V., 2017. Software-defined networking for internet of things: A survey. IEEE Internet of Things Journal 4, 1994–2008. DOI:10.1109/JIOT.2017.2746186.

Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O'Connor, B., Radoslavov, P., Snow, W., Parulkar, G., 2014. Onos: towards an open, distributed sdn os. HotSDN 2014 - Proceedings of the ACM SIGCOMM 2014 Workshop on Hot Topics in Software Defined Networking DOI:10.1145/2620728.2620744.

Biscani, F., Izzo, D., 2020. A parallel global multiobjective framework for optimization: pagmo. Journal of Open Source Software 5, 2338.

Booth, H., Rike, D., Witte, G., 2013. The national vulnerability database (nvd): Overview. Technical Report. National Institute of Standards and Technology.

Chehida, S., Baouya, A., Bozga, M., Bensalem, S., 2020. Exploration of impactful countermeasures on iot attacks, in: 2020 9th Mediterranean Conference on Embedded Computing (MECO), pp. 1–4. DOI:10.1109/MECO49872.2020.9134200.

Chung, C.j., Member, S., Khatkar, P., Xing, T., 2013. NICE : Network Intrusion Detection and Countermeasure. IEEE Transactions on Dependable Secure Computting 10, 1–14.

Correa Chica, J.C., Imbachi, J.C., Botero Vega, J.F., 2020. Security in sdn: A comprehensive survey. Journal of Network and Computer Applications 159, 102595. DOI:https://DOI.org/10.1016/j.jnca.2020.102595.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197. DOI:10.1109/4235.996017.

Dewri, R., Ray, I., Poolsappasit, N., Whitley, D., 2012. Optimal security hardening on attack tree models of networks : a cost-benefit analysis. International Journal of Information Security volume , 167–188 DOI:10.1007/s10207-012-0160-y.

Doynikova, E., Kotenko, I., 2018. The Multi-Layer Graph Based Technique for Proactive Automatic Response Against Cyber Attacks. Proceedings - 26th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2018 , 470–477DOI:10.1109/PDP2018.2018.00081.

Emmerich, M.T., Deutz, A.H., 2018. A tutorial on multiobjective optimization: Fundamentals and evolutionary methods. Natural Computing: An International Journal 17, 585–609. URL: https://DOI.org/10.1007/s11047-018-9685-y.

Enoch, S.Y., Hong, J.B., Ge, M., Khan, K.M., Kim, D.S., 2019. MultiObjective Security Hardening Optimisation for Dynamic Networks. ICC 2019 - 2019 IEEE International Conference on Communications (ICC) , 1–7.

Fan, Y.J., Yin, Y.H., Xu, L.D., Zeng, Y., Wu, F., 2014. Iot-based smart rehabilitation system. IEEE Transactions on Industrial Informatics 10, 1568–1577. DOI:10.1109/TII.2014.2302583.

Foremski, P., Nowak, S., Fröhlich, P., Hernández-Ramos, J.L., Baldini, G., 2020. Atopolicy: Automated traffic policing for improved iot network security. Sensors 20.

Foundation, O.N., .Openflow switch specification. URL: https://opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v13.0.pdf.

Garcia-Alfaro, G.G.G.J.R.H.J., 2017. Using an Event Data Taxonomy to Represent the Impact of Cyber Events as Geometrical Instances. IEEE Access 6, 8810–8828. DOI:10.1109/ACCESS.2017.2740402.

Garzia, F., Lombardi, M., Ramalingam, S., 2017. An integrated internet of everything - Genetic algorithms controller - Artificial neural networks framework for security/safety systems management and support. Proceedings - International Carnahan Conference on Security Technology 2017-Octob, 1–6. DOI:10.1109/CCST.2017.8167863.

Gelenbe, E., Domanska, J., Fröhlich, P., Nowak, M.P., Nowak, S., 2020. Self-aware networks that optimize security, qos, and energy. Proceedings of the IEEE 108, 1150–1167. DOI:10.1109/JPROC.2020.2992559.

Gonzalez-Granadillo, G., Garcia-Alfaro, J., Alvarez, E., El-Barbori, M., Debar, H., 2015. Selecting optimal countermeasures for attacks against critical systems using the attack volume model and the RORI index. Computers and Electrical Engineering 47, 13–34. DOI:10.1016/j.compeleceng.2015.07.023.

Gonzalez-Granadillo, G., Garcia-Alfaro, J., Debar, H., 2017. A polytopebased approach to measure the impact of events against critical infrastructures. Journal of Computer and System Sciences 83, 3– 21. DOI:10.1016/j.jcss.2016.02.004.

Granadillo, G.G., Débar, H., Jacob, G., Gaber, C., Achemlal, M., 2012. Individual countermeasure selection based on the return on response investment index, in: International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security, Springer. pp. 156–170.

Group, F.C.S.I., 2019. Common Vulnerability Scoring System version 3.1 Specification Document. Standard. FIRST-Forum of Incident Response and Security Teams. Cary, USA.

Gu, S., Hao, T., 2018. A pointer network based deep learning algorithm for 0–1 knapsack problem, in: 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI), pp. 473–477. DOI:10.1109/ICACI.2018.8377505.

Han, R., Gao, Y., Wu, C., Lu, D., 2018. An effective multi-objective optimization algorithm for spectrum allocations in the cognitive-radio based internet of things. IEEE Access 6, 12858–12867.

Hasan, K., Shetty, S., Hassanzadeh, A., Salem, M.B., Chen, J., 2018. ModelingCostofCountermeasuresinSoftwareDefinedNetworking-enabled Energy Delivery Systems. 2018 IEEE Conference on Communications and Network Security (CNS) , 1–9DOI:10.1109/CNS.2018.8433172.

Hildmann, H., Atia, D.Y., Ruta, D., Poon, K., Isakovic, A.F., 2019. Nature-Inspired Optimization in the Era of IoT: Particle Swarm Optimization (PSO) Applied to Indoor-Distributed Antenna Systems (I-DAS). Springer International Publishing, Cham. chapter 3. p. 171–192. DOI:10.1007/978-3-319-93100-5_11.

Huang, J., Xu, L., Xing, C.c., Duan, Q., 2015. A novel bioinspired multiobjective optimization algorithm for designing wireless sensor networks in the internet of things. Journal of Sensors 2015.

Kalamaras,I.,Drosou,A.,Tzovaras,D.,2014. Multi-objective optimization for multimodal visualization. IEEE transactions on multimedia 16, 1460–1472.

Kaul, S., Kumar, Y., Ghosh, U., Alnumay, W., 2021. Nature-inspired optimization algorithms for different computing systems: novel

perspective and systematic review. Multimedia Tools and Applications DOI:10.1007/ s11042-021-11011-x.

Khadr, M.H., Salameh, H.B., Ayyash, M., Almajali, S., Elgala, H., 2019. Securing iot delay-sensitive communications with opportunistic parallel transmission capability, in: 2019 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. DOI:10.1109/GLOBECOM38437.2019.9013884

Khan, I.U., Qureshi, I.M., Aziz, M.A., Cheema, T.A., Shah, S.B.H., 2020. Smart iot control-based nature inspired energy efficient routing protocol for flying ad hoc network (fanet). IEEE Access 8, 56371–56378. DOI:10.1109/ACCESS.2020.2981531.

Kheir, N., Cuppens-Boulahia, N., Cuppens, F., Debar, H., 2010. A service dependency model for cost-sensitive intrusion response, in: Proceedings of the 15th European Conference on Research in Computer Security, Springer-Verlag, Berlin, Heidelberg. p. 626–642.

Kotenko, I., Doynikova, E., 2016. Dynamical Calculation of Security Metrics for Countermeasure Selection in Computer Networks. Proceedings - 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, PDP 2016 , 558–565 DOI:10.1109/PDP.2016.96.

Lai, Y.H., Wu, T.C., Lai, C.F., Yang, L.T., Zhou, X., 2021. Cognitive optimal-setting control of aiot industrial applications with deep reinforcement learning. IEEE Transactions on Industrial Informatics 17, 2116–2123. DOI:10.1109/TII.2020.2986501.

Latah, M., Toker, L., 2019. Artificial intelligence enabled software-defined networking: a comprehensive overview. IET Networks 8, 79–99. DOI:https://DOI.org/10.1049/iet-net.2018.5082, 133–167.

Pal, S., Hitchens, M., Rabehaja, T., Mukhopadhyay, S., 2020. Security requirements for the internet of things: A systematic approach. Sensors 20. URL: DOI:10.3390/ s20205897.

Li, F., Li, Y., Yang, Z., Guo, Y., Yin, L., Wang, Z., 2018. Selecting combined countermeasures for multi-attack paths in intrusion response system, in: 2018 27th International Conference on Computer Communication and Networks (ICCCN), pp. 1–9. DOI:10.1109/ICCCN.2018.

Liu, Q., Li, X., Liu, H., Guo, Z., 2020. Multi-objective metaheuristics for discrete optimization problems: A review of the stateof-the-art. Applied Soft Computing 93, 106382. DOI:https://DOI.org/10.1016/j.asoc.2020.106382.

Mahdavi, M., Fesanghary, M., Damangir, E., 2007. An improved harmony search algorithm for solving optimization problems. Applied Mathematics and Computation 188, 1567 – 1579. DOI:https://DOI.org/10.1016/j.amc.2006.11.033. Mahmoud, R., Yousuf, T., Aloul, F., Zualkernan, I., 2015. Internet of things (IoT) security: Current status, challenges and prospective measures, in: 2015 10th International Conference for Internet Technology and Secured Transactions (ICITST), pp. 336–341. DOI:10.1109/ICITST.2015.7412116.

Mell,P.,Scarfone,K.,Romanosky,S.,2006. Common vulnerability scoring system. IEEE Security & Privacy 4, 85–89.

Messac, A., Ismail-Yahaya, A., Mattson, C., 2003. The normalized normal constraint method for generating the pareto frontier. Structural and multidisciplinary optimization 25, 86–98.

Miehling, E., Rasouli, M., Teneketzis, D., 2015. Optimal Defense Policies for Partially Observable Spreading Processes on Bayesian Attack Graphs. MTD '15: Proceedings of the Second ACM Workshop on Moving Target Defense , 67–76DOI:10.1145/2808475.2808482.

Ouyang, H.b., Gao, L.q., Li, S., Kong, X.y., Wang, Q., Zou, D.x., 2017. Improved harmony search algorithm. Appl. Soft Comput. 53, gateway. IEEE Access 8, 3159–3170. DOI:10.1109/ACCESS.2019.2960508.

Papachristou, K., Theodorou, T., Papadopoulos, S., Protogerou, A., Drosou, A., Tzovaras, D., 2019. Runtime and routing security policy verification for enhanced quality of service of iot networks, in: 2019 Global IoT

Lee, Y., Choi, T.J., Ahn, C.W., 2017. Multi-objective evolutionary approach to select security solutions. CAAI Transactions on Intelligence Technology 2, 64–67. Summit (GIoTS), pp. 1–6. DOI:10.1109/GIOTS.2019.8766404.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: An imperative style, high-performance deep learning library, in: Advances in neural information processing systems, pp. 8026–8037.

Pescador-Rojas, M., Coello, C.A.C., 2018. Studying the effect of techniques to generate reference vectors in many-objective optimization, in: Proceedings of the Genetic and Evolutionary Computation Conference Companion, Association for Computing Machinery, New York, NY, USA. p. 193–194. DOI:10.1145/3205651.3205684.

Poolsappasit, N., Dewri, R., Ray, I., 2012. Dynamic security risk management using Bayesian attack graphs. IEEE Transactions on Dependable and Secure Computing 9, 61–74. DOI:10.1109/TDSC.2011.34.

Prasanth, A., Jayachitra, S., 2020. A novel multi-objective optimization strategy for enhancing quality of service in iot-enabled wsn applications. Peer-to-Peer Networking and Applications 13, 1905–1920.

Protogerou, A., Papadopoulos, S., Drosou, A., Tzovaras, D., Refanidis, I., 2020. A graph neural network method for distributed anomaly detection in iot. Evolving Systems , 1–18.

Rachedi, A., Benslimane, A., 2016. Multi-objective optimization for security and qos adaptation in wireless sensor networks, in: 2016 IEEE International Conference on Communications (ICC), pp. 1–7. DOI:10.1109/ICC.2016.7510879.

Ramírez, P.L.G., Taha, M., Lloret, J., Tomás, J., 2020. An intelligent Algorithm for resource sharing and self-management of wireless-iot. arXiv: https://ietresearch.onlinelibrary.wiley.com/DOI/pdf/10.1049/iet-net.2018.508

Ravì, D., Wong, C., Lo, B., Yang, G., 2017. A deep learning approach to on-node sensor data analytics for mobile or wearable devices. IEEE Journal of Biomedical and Health Informatics 21, 56–64. DOI:10.1109/JBHI.2016.2633287.

Riquelme N., Von Lucken C. and Baran B., 2015. Performance Metrics in mult-objective optimization. Latin American Computing Conference (CLEI) 1-11, doi: 10.1109/CLEI.2015.7360024.

Rontidis, G., Panaousis, E., Laszka, A., Dagiuklas, T., Malacaria, P., Alpcan, T., 2015. A game-theoretic approach for minimizing security risks in the internet-of-things, in: 2015 IEEE International Conference on Communication Workshop (ICCW), pp. 2639–2644. DOI:10.1109/ICCW.2015.7247577.

Roy, A., Kim, D.S., Trivedi, K.S., 2012. Scalable Optimal Countermeasure Selection using Implicit Enumeration on Attack Countermeasure Trees. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN 2012) .

Salman, O., Elhajj, I.H., Kayssi, A., Chehab, A., 2016. Sdn controllers: A comparative study, in: 2016 18th Mediterranean Electrotechnical Conference (MELECON), pp. 1–6. DOI:10.1109/MELCON.2016.7495430.

Shameli-Sendi, A., Dagenais, M., 2015. ORCEF: Online response cost evaluation framework for intrusion response system. Journal of Network and Computer Applications 55, 89–107. DOI:10.1016/j.jnca.2015.05.004.

Shameli-Sendi, A., Louafi, H., He, W., Cheriet, M., 2018. Dynamic optimal countermeasure selection for intrusion response system. IEEE Transactions on Dependable and Secure Computing 15, 755–770. DOI:10.1109/TDSC.2016.2615622.

Siddique, N., Adeli, H., 2013. Computational intelligence: synergies of fuzzy logic, neural networks and evolutionary computing. John Wiley & Sons.

Sloss, A.N., Gustafson, S., 2020. 2019 evolutionary algorithms review, in: Genetic Programming Theory and Practice XVII. Springer, pp. 307– 344.

Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks. Advances in neural information processing systems 27, 3104–3112.

Tahsien, S.M., Karimipour, H., Spachos, P., 2020. Machine learning based solutions for security of internet of things (iot): A survey. Journal of Network and Computer Applications 161,102630. DOI:https://DOI.org/10.1016/j.jnca.2020.102630.

Toldinas, J., Venckauskas, A., Damasevicius, R., Grigaliunas, S., Morkevicius, N., Baranauskas, E., 2021. A novel approach for network intrusion detection using multistage deep learning image recognition. Electronics 10. DOI:10.3390/ electronics10151854.

. CYBER; Methods and protocols; Part 1: Method and pro forma for Threat, Vulnerability, Risk Analysis (TVRA). Standard. ETSI. Sophia Antipolis Cedex, FR.

Varga, P., Kathareios, G., Máté, Á., Clauberg, R., Anghel, A., Orosz, P., Nagy, B., Tóthfalusi, T., Kovács, L., Gusat, M., 2018. Real-time security services for SDN-based datacenters. 2017 13th International Conference on Network and Service Management, CNSM 2017 2018-Janua, 1–9. DOI:10.23919/CNSM.2017.8256030.

Viduto, V., Maple, C., Huang, W., López-Peréz, D., 2012. A novel Risk Assessment and Optimisation Model for a multi-objective network security countermeasure selection problem. Decision Support Systems 53, 599–610. DOI:10.1016/j.dss.2012.04.001.

Vinyals, O., Fortunato, M., Jaitly, N., 2015. Pointer networks, in: Advances in Neural Information Processing Systems 28, pp. 2692–2700.

Wang, S., Zhang, Z., Kadobayashi, Y., 2013. Exploring attack graph for cost-benefit security hardening: A probabilistic approach. Computers and Security 32, 158–169. DOI:10.1016/j.cose.2012.09.013.

Xu, X., Fu, S., Qi, L., Zhang, X., Liu, Q., He, Q., Li, S., 2018. An iot-oriented data placement method with privacy preservation in cloud environment. Journal of Network and Computer Applications 124,148–157. DOI:https://DOI.org/10.1016/j.jnca.2018.09.006.

Yan, Z., Susilo, W., Bertino, E., Zhang, J., Yang, L.T., 2020. Ai-driven data security and privacy. Journal of Network and Computer Applications 172, 102842. DOI:https://DOI.org/10.1016/j.jnca.2020.102842.

Zhang, J., Tao, D., 2020. Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things. IEEE Internet of Things Journal , 1–1DOI:10.1109/JIOT.2020.3039359.

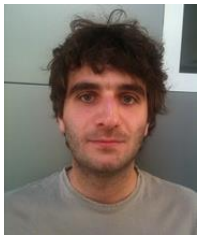
Zhang, Q., Li, H., 2007. MOEAD/d: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11, 712–731. DOI:10.1109/TEVC.2007.892759.

Zonouz, S., Haghani, P., 2013. Cyber-physical security metric inference in smart grid critical infrastructures based on system administrators' responsive behavior. Computers and Security 39, 190–200. DOI:10.1016/j.cose.2013.07.003.

Zonouz, S.A., Khurana, H., Sanders, W.H., Yardley, T.M., 2009. Rre: A game-theoretic intrusion response and recovery engine, in: 2009 IEEE/IFIP International Conference on Dependable Systems Networks, pp. 439–448. DOI:10.1109/DSN.2009.5270307.

Asterios Mpatziakas graduated from the department of Mathematics of the Aristotle University of Thessaloniki on 2015 and on 2017 he acquired a MSc. on "Complex Systems and Networks" from the same University. He is currently a research associate with the Information Technologies Institute of the Centre for Research and Technology Hellas (CERTH). Among his research interests are artificial intelligence, machine learning and discrete optimization problems especially concerning network related applications.

Anastasios Drosou is a reseacher at the Information Technologies Institute at the Centre for Research and Technology Hellas. He received his Diploma in Electrical and Computer Engineering, with an expertise in Telecommunications, from Aristotle University of Thessaloniki, and an MSc. in Communication Electronics from the Technische Universitt Mnchen in 2004 and in 2007, respectively. He also holds a PhD in Signal and Image processing from the Imperial College London since 2013. His research interests include biometric security, visualization and network security, regarding identification. He serves as a regular reviewer for several technical journals and he is the coauthor of more than 90 scientific papers (journals, conferences and book chapters).



Stavros Papadopoulos received the Diploma degree in electrical and computer engineering from the Aristotle University of Thessaloniki in 2010, and the Ph.D. degree in graph analytics from the Intelligent Systems and Networks Group, Imperial College London. He is currently a Research Assistant with the Information Technologies Institute of the Centre for Research and Technology Hellas. His main research interests include information visualization, visual analytics, network security, anomaly detection in mobile and IP networks, and root cause analysis.

Dimitrios Tzovaras is a senior researcher (Grade A) and also the president of the Board.at the Centre for Research and Technology Hellas. He is also a visiting professor at Imperial College London. Prior to his current position, he was a senior researcher in 3D imaging at Aristotle University of Thessaloniki. His main research interests include virtual reality, assistive technologies, 3D data processing, and stereo and multiview image sequence coding. His involvement with those research areas has led to the coauthoring of more than 60 papers in refereed journals and more than 150 papers in international conferences.