



LJMU Research Online

Lever, KE and Kifayat, K

Identifying and mitigating security risks for secure and robust NGI networks

<http://researchonline.ljmu.ac.uk/id/eprint/12922/>

Article

Citation (please note it is advisable to refer to the publisher's version if you intend to cite from this work)

Lever, KE and Kifayat, K (2020) Identifying and mitigating security risks for secure and robust NGI networks. Sustainable Cities and Society, 59. ISSN 2210-6707

LJMU has developed **LJMU Research Online** for users to access the research output of the University more effectively. Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Users may download and/or print one copy of any article(s) in LJMU Research Online to facilitate their private study or for non-commercial research. You may not engage in further distribution of the material or use it for any profit-making activities or any commercial gain.

The version presented here may differ from the published version or from the version of the record. Please see the repository URL above for details on accessing the published version and note that access may require a subscription.

For more information please contact researchonline@ljmu.ac.uk

<http://researchonline.ljmu.ac.uk/>

Identifying and Mitigating Security Risks for Secure and Robust NGI Networks

Abstract

Smart city development is important to achieve sustainable cities and societies which help enhance urban services, reduce resource consumption and decrease overall cost. The incorporation of smart cities with the Internet has given us the Next Generation of Internet (NGI) where every smart device exploits the interconnected services and infrastructure of the Internet. The underlying structure of NGI is composed of large scale heterogeneous multilevel systems-of-systems (SoSs) where each system represents a sensor, mobile phone, computer or smart device.

Security and privacy is a fundamental requirement of NGI which is heavily dependent on the composition of services and connectivity of the underlying systems. Meaning any unsecure system can affect the security of the entire networked infrastructure/SoSs.

Therefore, it is important to analyse and understand the composition of different systems at different levels in NGI in order to identify and mitigate vulnerabilities. This paper proposes a solution to identify and mitigate vulnerabilities within multilevel SoSs, to enhance security without deploying additional security at endpoints, and quantify security levels of individual systems and the entire composed system. The solution was tested and evaluated using simulation and a network testbed. Results show that NGI security can be enhanced with better composition of systems.

Keywords: Next Generation Internet; Internet of Things; Wireless Sensor Networks; Optimisation; Network Security; Interoperability.

1. Introduction

The advancement in smart cities technologies undoubtedly affects peoples' lives and it is clear that in the future, it will influence the reshaping of our communities and societies. Smart cities are bringing economic benefits, efficient public utilities, improved transportation, safety, sustainability, smart infrastructure, smart health care and more effective data driven decision making. All these benefits have been achieved through use of

Note: Colour should be used for all figures in print.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Next Generation Internet (NGI) infrastructure where all components of smart cities are connected and networked together through local networks and the Internet. NGI is an example of a complex system, complex network or system-of-systems (SoSs). In general, it is a large scale dynamic system composed of a large number of subsystems, that exhibit both highly nonlinear deterministic and stochastic characteristics and that are regulated at different levels, which evolves with the passage of time and emerges with a new set of challenges.

The European Union NGI initiative has listed the following research challenges in NGI: Cybersecurity & Resilience, Trustworthy online Information Infrastructure, Online identities and Trust, Decentralize Powering, The right of Opt Out & Self-Govern, Data Sovereignty, Ethical AI and machine learning, A Diverse and Safe Internet, An Accessible and Open Internet, and Sustainable and Fair Infrastructure (Sestini et al., 2018). Also, the Federal Trade Commission Report on IoT highlights consumer privacy concerns and security risks, advising organisations to adopt best practices to address these problems. The report states that smart devices are responsible for collating vast amounts of both critical and personal data (Tariq et al., 2019), and data is not just at risk from unauthorised access it is vital to uphold data integrity. As data corruption can easily cause an object that is reliant upon that data to malfunction in unpredicted and dangerous ways.

Cybersecurity is a major problem in today's interconnected world, as there is a major theoretical and applied shortfall in current cybersecurity architecture (Walker-Roberts et al., 2018). In addition, it is getting more complex, scalable and the threat surface is getting more dynamic due to the expansion of the Internet's infrastructure, development of billions of new heterogeneous IoT devices, increasing interconnectedness, and the development of new software's such as operating systems and applications (Networld, 2020). This naturally creates a high dependency between systems at different hierarchy which raise the risks of massive breakdowns, either through an accidental glitch or a malicious attack. Similar concept applies to security solutions proposed for NGI i.e. an unsecure wrongly connected system can directly affect the security of interconnected subsystems (Ai et al., 2019) and overall NGI.

The security of NGI heavily depends on the composition of services (Meland, 2011, Papazoglou et al., 2006, Aniketos, 2011) and connectivity (Zhong-Yuan et al., 2019) of underlying systems. Many empirical studies (Zhong-Yuan et al., 2019, Albert et al., 2000, Jiang et al., 2018) have proved that important node set mining is very critical in a network, that is, a portion of vital nodes may lead to the collapse of the whole interdependent network (e.g. power grids and communication networks) (Jiang et al. 2018). It can also be used by IT infrastructure and service providers to control the Internet traffic on many critical nodes for virus search (Liu et al., 2011). Due to increases in the number of distinct interconnected devices, it has become difficult to develop a dynamic and reliable security solution that can safeguard the network against all potential security risks (Tariq et al., 2019). Therefore, it is important to analyse and understand the connectivity of different

systems at different levels in order to identify vulnerabilities and mitigate them (Holme et al., 2002, Carmi et al., Albert et al., 2000).

These vulnerabilities include cascading effect (Liu et al., 2011, Buldyrev et al., 2010), node removal (Lekha and Balakrishnan, 2018), vital link identification (Liang et al., 2017), controllability of nodes (Liu et al., 2011), iterative path attacks (Pua et al., 2015) and seeding strategies for large scale propagation (Hinz et al., 2011). Moreover, robustness analysis and mitigation of such vulnerabilities has been investigated in Chen et al. (2017), Wu et al. (2018), Jiang et al. (2016), Du et al. (2016), Liu et al. (2016), and Liu et al. (2011). However, these solutions may not be as effective as they focus on network connectivity and its features (centralities) only. There are many other important properties of the system, network and their security which are not considered and directly affect the performance and security of the entire system.

The presented solution in this paper, not only considers network connection properties, but also considers the actual properties of the system and network. This helps to identify vulnerable nodes more effectively. For example, IoT device architecture makes it difficult to embed security solutions on all devices (Tariq et al., 2019), therefore, it is vital to detect these nodes that have the potential to expose the entire infrastructure. An additional feature of the proposed solution is the quantification of the security levels of the subsystems and entire system (Section 4.1) which leads toward our third contribution, to find the best possible composition which will result in a secure system (Section 4.3). The robustness of the network can be more accurately analysed by accounting for weights in the system that in turn can make the entire system more resilient against different attacks.

This paper is organised as follows: Section 2 describes related work, examines the data access control problem for smart cities, and considers the problematic relational states between nodes. Section 3 presents the proposed solution and its functions. Section 4 provides details of the proposed solutions application, overview of the implementation process and the simulated environment, and provides initial results from the simulations. In Section 5 we conclude and discuss future work.

2. Related work

This section presents related research work on vulnerability analysis, important node identification and the composition of different services in complex networks or SOSs scenario. These three domains directly relate to the proposed solution. The important node ranking research and development has attracted a lot of attention in the past few decades. The famous PageRank algorithm performs link analysis of the large scale interconnected web using different network centralities (degree, closeness, and betweenness) to identify important nodes. This area has also been widely explored in physics and mathematics as it has many real world applications. NGI is considered to be an adaptive complex system or network which has many dynamic features and evolves with the passage of time. It naturally produces many attack vulnerabilities, e.g. Boccaletti et al. (2007) proposed a

solution to quantify the vulnerabilities of node and edge deletion of a multiscale complex network, Goldshtein et al. (2004) measures vulnerabilities in hierarchy of complex networks and Mishkovski et al. (2011) measure average edge betweenness vulnerability using a metric. Whereas Nie et al. (2015) proposed solution based two new attack strategies based on both degree and betweenness. Moreover, Lu et al. (2016) and Du et al. (2017) studied the identification of vital nodes in the network and concluded that adaptive recalculation strategies are more efficient in comparison to straightforward methods.

According to Sindhu et al. (2018), most real-world networks are weighted and vulnerability analysis of weighted networks is a new area of research. The efficiency of the attacks is directly proportion to the network weights (Bellingeri and Cassi, 2018). It means vulnerability analysis without considering the weighted structure of networks could end in misleading results. Cascading vulnerability (Cadini et al., 2017, Feng et al., 2017) is another important area researched in complex networks. Its identification is similar to vital node and link identification work (Lu et al., 2016, Du et al., 2017). However, it focuses on minimising the damage of vital node (Liu et al., 2011, Buldyrev et al., 2010, Wu et al., 2019, Brummitt et al., 2012, Cai et al., 2016).

Other vulnerabilities include node removal (Sindhu et al., 2018, Boccaletti et al., 2007), controllability of nodes (Liu et al., 2011, Hinz et al, 2011) with suitable choice of seeding strategies for large scale propagation. Moreover, robustness analysis and mitigation of such vulnerabilities has been investigated in Chen et al. (2017), Wu et al. (2018), Jiang et al. (2016), Du et al. (2016), Liu et al. (2016), Liu et al. (2011), and Aia et al. (2019).

The zero trust security model has also been explored in regards to the complicated future internet. Walker-Roberts et al. (2018) review existing methods in order to ascertain existing technological capabilities that can mitigate insider threats within cybersecurity systems, identifying that most security measures require a breach to occur before the threat can be analysed and future malicious activity prevented. This type of reactive approach will not be effective to protect the future NGI, as a single breach has the potential to result in catastrophic events occurring or failures cascading across to other networked infrastructure that could impede collaborative systems. Belguith et al. (2018) aim to address high level security issues in today's internet, proposing a cryptographic method to assure cooperative data aggregation based on the use of an attribute based signcryption scheme. The proposed method aims to ensure that data collated via IoT devices can be authenticated and that all devices will be protected from unauthorised access.

Considering that some variants have existing common vulnerabilities and focusing on the vulnerability-aware diverse variants deployment problem as an integer-programming problem, Jianjian et al. (2019) propose the Vulnerability-aware Heterogeneous Network Devices Assignment (VHNDA) and outline their Simulated Annealing Vulnerability-aware Diverse Variants Deployment (SA-VDVD) method and present a low complexity algorithm named Graph Segmentation-based Simulated Annealing Vulnerability-aware Diverse Variants Deployment (GSSA-VDVD). In an attempt to prevent the spread of malicious

packets. Effectively preventing the spread of malicious packet attacks compared to some baseline algorithms.

Considering that some variants have existing common vulnerabilities and focusing on the vulnerability-aware diverse variants deployment problem as an integer-programming problem. Jianjian et al. (2019) propose a quantitative metric that prevents the spread of malicious packets, and outline a low complexity algorithm named Graph Segmentation-based Simulated Annealing Vulnerability-aware Diverse Variants Deployment (GSSA-VDVD). In order to address high computational complexity as network sizes increase. Tariq et al. (2019) survey the challenges of securing future digital infrastructures as they continuously evolve, and analyse the cybersecurity challenges, big data privacy and trust concerns associated with fog-enabled IoT.

Meland (2011) explained the importance of service composition in future internet. Where users can mix, match and create rapid-growth services. However, all this comes with a new set of security challenges. This work has been investigated in detail in the major EU FP7 funded project ANIKETOS (2017). Stanford University Security Lab (Datta et al., 2004) used a similar approach, using different security components to compose a secure protocol. It has also been highlighted that information revealed by one component may interfere with the security of the other. A similar idea has been applied to systems and software, Alfarhan and Alsohaily (2017) critically analyse self-organising networks, consider long-term evolution systems, and identify several network parameter optimisation challenges associated with the development of these types of network. Proposing a Mixed Integer Quadratic Program optimisation technique for each of the identified challenges (optimisation of frequency channel assignments, tracking area codes, physical cell identifiers, and long-term evolution). Whereas, Yao, et al. (2017) only simulate a small sized network graph.

There is a severe theoretical deficit in cybersecurity architecture, and while many aspects of research and applied cybersecurity solutions attempt to address high level security issues in today's internet and the NGI, numerous solutions only consider a single aspect of network vulnerabilities which is not effective when it comes to the security of the overall system. The proposed solution in this paper considers the composition between multiple elements and levels which add more complexity. It identifies vulnerable nodes in the context of its connectivity and measures local security properties and the composed system, which is different and not considered by any of the other approaches described in this section. Finally, it tries all possible combinations in order to find the best possible configuration of node connectivity and security properties.

3. Security Risk Analysis and Mitigation (SCRAM) framework

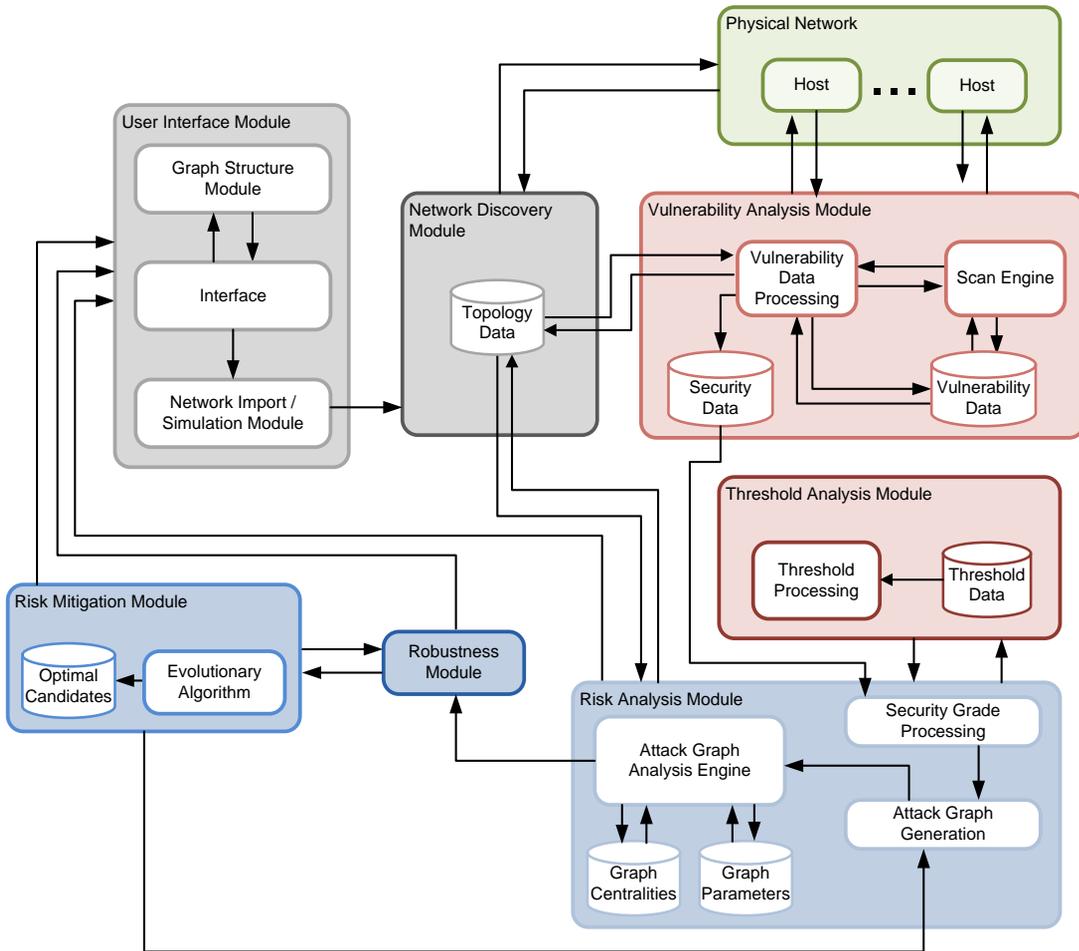


Fig. 1. Illustrated Overview of the SCRAM Framework

The Security Risk Analysis and Mitigation (SCRAM) methodology is an extension of the Engineering and Physical Sciences Research Council Project (EPSRC) developed to secure component composition for personal ubiquitous computing (EPSRC 2017) and the European Union (FP7-ICT) funded project ANIKETOS (ANIKETOS, 2017). SCRAM has the ability to either generate a random network or import an existing network topology. The SCRAM framework is presented in Figure 1. The proposed solution simulates a subpart of an NGI environment, which demonstrates the feasibility of the techniques proposed. By simulating a small sector, graph and data sets are reduced allowing us to intuitively analyse results, and assess the framework and algorithms effectiveness. SCRAM components are as follows:

User Interface Module: The User Interface Module allows security managers to utilise the Network Import Module to either import an existing network into the framework for vulnerability analysis and risk mitigation, or allows for a network to be simulated based on the selected parameters. This can assist with the design and development of ICT infrastructures by simulating networked systems, then analysing and reconfiguring the networks to mitigate risks and increase security. The interface allows for a single network/SoS infrastructure to be selected and developed for analysis or can initiate multi-level SoS infrastructures (NGI) for evaluation, via the Network Import Module.

Within the interface module the *Graph Structure Module* allows the user to select the graph structure type for security optimisation and risk mitigation, including the parameters for prioritisation during the risk mitigation process. For instance, if users wish to improve security and mitigate risks focusing on the networks node security grades and robustness, they will not wish to select the graph structure which prioritises and visualises node energy efficiency during the risk mitigation process.

Network Discovery Module: The *Network Discovery Module* is an automated process that systematically discovers networked devices and assists to map devices identified and their communication links within the *Physical Network* infrastructure, including devices and systems which share a collaborative relationship. Producing a detailed inventory which includes device type, operating system, whether encryption, firewalls, and intrusion detection systems are utilised, if anti-virus and security software is installed on the nodes, if the device has internet access, and the assigned data access for the node, etc. This information is stored within the *Topology Data* database, which can be accessed and utilised by both the *Vulnerability Analysis Module* and *Risk Analysis Module*.

Vulnerability Analysis Module: This module accesses the Topology Data database via the Vulnerability Data Processing unit, which is responsible for determining the appropriate vulnerability scans for each node that has been identified as unscanned or its scan is considered outdated. Once the necessary scans have been conducted utilising the *Scan Engine* unit, *Vulnerability Data* database and utilising the topology data, the *Vulnerability Data Processing* unit will assess the networks nodes and evaluate the risks, recording the findings and updating information as necessary in the *Vulnerability Data* and *Security Data* databases.

Risk assessment methodologies when applied to networks directly can impact the functionality of some systems and their components. Therefore the *Vulnerability Analysis Module* will identify the nodes which are unable to be scanned for vulnerabilities, and the risk that these unscanned nodes pose to the network/SoS will be quantified as part of the vulnerability analysis. The vulnerability scoring and exploit databases currently incorporated into the SCRAM frameworks *Vulnerability Analysis Module* are examined in Section 4.2.

Risk Analysis Module: This module serves several purposes; firstly the security data for each node is passed from the *Security Data* database to the *Security Grade Processing* unit. This unit is responsible for quantifying each nodes security grade based on the findings of the vulnerability analysis, these grades will then be compared to the relevant thresholds as part of the risk analysis process, and will be utilised as part of the attack graph generation method to assist with visualising node status. Security grade assignment is discussed in detail in Section 4.2. The *Attack Graph Generation* unit within this module utilises the updated topology data stored in the *Topology Data* database, threshold analysis data stored in the *Threshold Data* database, and the quantified security grades to generate an attack graph which will help establish a visualised representation of the network topology, security status, and data access violations, or can visualise nodes based on energy efficiency levels depending on the graph structure selected. After the evolutionary risk mitigation process has been implemented, the *Attack Graph Generation Module* will also be used to generate the improved optimal candidate graphs. The *Attack Graph Analysis Engine* evaluates each graph that has been generated, quantifying both network centralities (described in section 3.2) and node centralities, with the results being stored within the *Graph Centralities* database.

Threshold Analysis module: This module is primarily used by *Threshold Processing* unit to identify data access violations and node security status. The thresholds will be established by the network security managers and these profiles will be stored within the *Threshold Data* database. During the risk analysis stage, as security grades are assigned to each node, for example, the *Risk Analysis Module* will pass these grades onto the Threshold Analysis Module for assessment, with results being stored within the *Threshold Data* database. The *Security Grade Processing* unit will then pass on the assessed results to the *Attack Graph Generation* unit which incorporates these results into the graph to ensure that insecure nodes and data access violations can be intuitively identified.

Robustness Module: This module is responsible for measuring each node within the network by means of a robustness function after an attack graph has been generated and analysed. During the risk mitigation process the *Robustness Module* will quantify the robustness of each node based on five key parameters which have been generated by the *Risk Analysis Module*; this method is described in detail in Section 4.3 **Error! Reference source not found.** An overall robustness level is then quantified for the network, and during the risk mitigation process this level assists the evolutionary algorithm to produce a new generation of improved solutions. The robustness score of the network also is of great benefit as it provides an assigned numerical value to the entire network to establish its appropriateness, and can be used as a comparative evaluation number as improvements are made to the security of the network/SoS.

Risk Mitigation Module: This module contains an *Evolutionary Algorithm*, to overcome the limitations of local search techniques in large complex networks. Utilising key parameters generated by both the *Risk Analysis Module* and *Robustness Module*, this process generates a new set of potential solutions which are then evolved for comparison, in order to find a set of best solutions. Inadequate solutions die out as they are replaced with new better identified solutions. Each solution is fully analysed via the *Risk Analysis Module* and *Robustness Module* to ensure that only the best individuals are directly passed to the next generation of solutions until the end criteria is met. Improved solutions are stored within the *Optimal Candidates* database, and will be passed to the *User Interface Module* to allow for the generated undirected graphs, combined with the reports generated by the *Risk Analysis Module* to be critically assessed by the security managers and decision makers.

3.1. Data access control for smart city scenario

NGI are composed of many different types of networks (System-of-Systems). Smart city infrastructure is a major part and a large network inside the NGI. In this paper we, assume smart cities infrastructure which relies upon the generation and distribution of data, the security of this data and access control is problematic. In part, due to the large generation and exchange of confidential and security sensitive data across a vast amount of distinct devices (Tariq et al., 2019, Walker-Roberts et al., 2018). Generated data could include sensitive data such as an individual's location, personal or professional information, state of health, life events, habits, etc. (Belguith et al., 2018).

A demonstrative example scenario based on a subpart of the NGI or smart city is provided in Figure 2, depicting communications between emergency services (mobile devices), transportation (sensor), and the local government (server) in response to an emergency. Collaborative nodes are connected via varying communication links that include smart devices and a static sensor, with differing security levels.

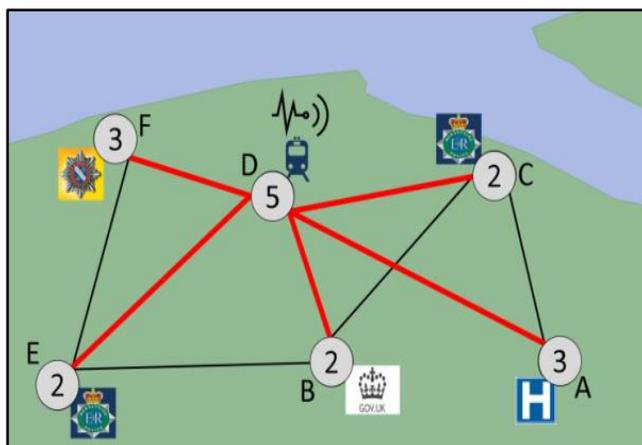


Fig. 2. Composed data access control scenario of a smart city

Unencrypted data with a security level of 3 is being forwarded across the collaborative network, between A and F. Figure 2 visualises every possible secure and unsecure connection in which data with the appropriate security level can traverse between the two nodes, with thick red lines indicating data access violations.

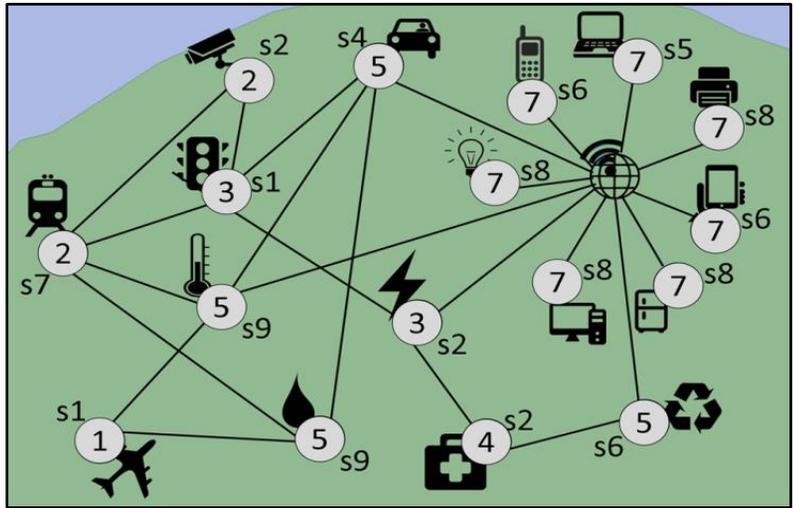


Fig. 3. Composed data access control Scenario 2, consisting of sensitivity levels and data flow

We assume that many smart cities consist of a large number of IoT devices, that include physical devices such as actuators, sensors, smart phones, personal digital assistants (PDAs), radio-frequency identification (RFID) tags, wearable smart devices (e.g. health care monitors, smart watches), smart meters or any ‘thing’ with embedded software (Tariq et al., 2019). When we can overview the topology of such collaborative networks, we see that they are a combination of diverse, smart and resource constrained devices (Tariq et al., 2019, Belguith et al., 2018) that sense data or control and interact with other systems and objects.

This type of topology could form a complex series of differing communication links across a city, with devices connecting and transferring different types of data, in a variety of formats, and via various protocols. The majority of IoT devices used within smart cities are developed and distributed by different manufactures, and the security of these devices can lack any form of accepted industry standards. In addition, organisations also have different security frameworks and their own standards (Tariq et al., 2019), this means securing these devices and collaborative infrastructures is problematic and interoperability is more complex. The scenario in Figure 3 is a demonstrative example of such devices and connections. In this scenario we cannot simply block and reroute data via different communication paths to devices, as the IoT objects are developed to connect, access data or control other devices. Moreover, devices with lower security and data access levels may be

required to interact with devices that have higher security and data access levels within the city.

Recognising the significance of the data access control problem as surveyed in (Zhou et al., 2008) which outlines a principal model of access control (MATTS), using these principal concepts and building upon previous solutions using the MATTS tool to identify such vulnerabilities within crisis management scenarios, we have evolved the model and propose a new solution to compose data security and improve the data flow security of the overall network, which we convey in Sections 3 and 4.

3.2. Network centralities

Table 1. Node centrality indicators and their respective equations

Centrality	Description	Equation
Degree (Freeman, 1979)	Identifies how popular or active a node is within a network, higher degree values indicate a nodes dominance within the network. Where $deg(u)$ is the number of node u 's edges and V is the set of nodes in the network.	$C_{deg}(u) = \frac{deg(u)}{ V -1} \quad (1)$
Betweenness (Freeman, 1977)	Nodes situated on the shortest path route are often the nodes most relied upon to transfer data. High betweenness values indicate a nodes importance in regards to data flow, and can determine single points of failure in environments. where $\sigma_{s,t}$ is the total number of shortest paths from source node s to destination node t , and $\sigma_{s,t}(u)$ is the number of shortest paths from source node s to destination node t which actually pass through node u .	$C_{bet}(u) = \frac{1}{(V -1) \cdot (V -2)} \sum_{s \neq u, t \neq u \in V} \frac{\sigma_{s,t}(u)}{\sigma_{s,t}} \quad (2)$
Closeness (Sabidussi, 1966)	Identifies nodes with the shortest path, and those which are uniquely accessible to all nodes within the environment either directly or indirectly. Highly centralised networks are generally unstable, while low centralised networks in general are not prone to single points of failure. Where $dist(u,v)$ is the length of the shortest path from node u to node v .	$C_{clo}(u) = \frac{[\sum_{v \neq u \in V} dist(u,v)]^{-1}}{ V -1} \quad (3)$
Eigenvector (Freeman, 1979)	Identifies nodes which play a more prominent role within the network. This centrality is considered more advanced than degree centrality and it differentiates links that are not equal to each other. Where $N(u)$ is the set of nodes reachable directly from u and λ is a constant. With vector-matrix notation, this equation can be rewritten as $\lambda \cdot C_{eig} = W \cdot C_{eig}$ Where $C_{eig} = (C_{eig}(v))_{v \in V}$ and $W = (W_{u,v})_{u,v \in V}$. Therefore C_{eig} is an eigenvector of the weighted adjacency matrix W with eigenvalue λ .	$C_{eig}(u) = \frac{1}{\lambda} \sum_{v \in N(u)} W_{u,v} \cdot C_{eig}(v) \quad (4)$
Bridging (Hwang et al., 2016)	Identifies nodes that are densely connecting other nodes within a network, and whether the nodes topological location and data flow are reliant upon. Bridging centrality is accomplished by quantifying the networks betweenness centrality C_B and the bridging coefficient BC , thus measures a nodes global and local features. The bridging centrality $C_R(v)$ for v of interest is defined.	$C_R(v) = BC(v) \cdot C_B(v) \quad (5)$

In addition to the data access control problem we also consider the problematic relational states between nodes, in an attempt to identify vulnerabilities and critical risks which have the ability to expose NGI networks. Realised through the use of mathematical formulas and assignment of numeric numbers to risks, allowing for identified risks to be quantified and network topologies to be visualised. With advancements in the fields of graph theory, network theory and social network analysis, there has been considerable progress with mathematical and computational tools. This allows for important relationships between nodes to be conveyed, and can assist with ascertaining network behaviour characteristics. For instance, centrality indicators (degree, betweenness, closeness, eigenvector, and bridging) help to assist us with ascertaining a nodes (vertices) importance within a network (Kim and Song, 2013, Meghanathan, 2016), summarised in Table 1.

3.3. Energy efficiency

Although IoT devices or any device in the NGI can have a regular power supply within a smart city environment, it is strongly recommended and an NGI requirement as highlighted in the EU NGI project, that any proposed solution should be energy efficient. Typically, IoT is a combination of interconnected, diverse, smart and resource constrained devices that provide advanced services through the exchange of data. IoT applications are also deployed as Low power and Lossy Networks (LLN), e.g. as wireless sensor networks (WSN), smart city and smart health applications. This class of networks is also resource constrained, has high loss rates, low data rates and volatile communication links (Tariq et al., 2019, Belguith et al., 2018).

Organisations continually have concerns about the efficiency of their devices and networks, including how to reduce operational costs (Tariq et al., 2019). By reducing the amount of data transmitted via resource constrained and insecure networks and objects, we can enhance security, save energy, while reducing operational overheads (Belguith et al., 2018). IoT devices are also prone to failure due to environmental factors, which can cause changes to the resource strained network topology and impact the energy consumption of remaining nodes (Gao et al. 2002, Qiu et al., 2017).

4. Systems security composition

The proposed solution considers high risk nodes within the networked topology throughout the optimisation process, focusing upon nodes with a high degree of connectivity (i.e. nodes measured through bridging centrality). Nodes with high bridging centrality pose a great threat to networks, as should these nodes be compromised or a failure occurs, the impact caused to these critical points has the capacity to interrupt data flow and reduce interoperability. To minimise these risks, we optimise the network connectivity by changing connections among the nodes in order to determine the most secure combination

of links. In optimisation different techniques are used to evaluate their performance in different scenarios.

In addition to security factors (degree, bridging centrality and communication security level) and energy efficiency, we examine two natural factors during the optimisation process. These are average minimum path length, which takes the average of all shortest paths between pairs of nodes within the network, and the cost of communications. This is the sum of all link weights, calculated as the geodesic distance between connected nodes.

4.1. Data security level of the network

To determine the data security level of the network, we assume that the nodes which form the topology are static, yet have dynamic connectivity (i.e. nodes can change communication links). Each node within the NGI environment will be assigned an authorisation level, using the principal concepts surveyed in (Zhou et al., 2008) which we discussed in Section 2.2. In terms of security, it is vital that data is only passed via nodes along communication links with sufficient authorisation levels for that data flow. $S(N)$ represents the proportion of secure paths between pairs of nodes that are entitled to communicate.

$$S(N) = \frac{\sum_{\forall g \in G} \sum_{s,t \in V_g, s \neq t} \delta_{s,t}(g)}{\sum_{\forall g \in G} |V_g| \times (|V_g| - 1)} \quad (6)$$

In this equation G is the set of different grades that nodes inside the network N might have assigned, V_g is the set of nodes in the network that reach the required authorisation level to access the given data at level g , $\delta_{s,t}(g)$ is a step function taking the value 1 if it's possible to find a secure path between s and t , given the sensitivity level g and 0 otherwise; and $n=|N|$ is the number of nodes within the network.

4.2. Node security grade assignment

Typically, vulnerabilities are initially identified using a network vulnerability scanner, which provides an automated approach for hosts and network topology to be scanned. Popular scanners include Nessus, Retina, Nmap, Nesspose and MaxPatrol. These tools identify and provide data on vulnerabilities within the networks topology and hosts, generating details on weaknesses such as open ports, network configurations, system components, software applications and services, logins, and active IP addresses, etc.

Vulnerability scanners though must be used as part of a risk assessment strategy and not as a full standalone security solution, as they can struggle to identify vulnerabilities resulting in false positives. Unlike firewalls, anti-virus and intrusion detection systems,

vulnerability scanners provide a proactive approach to ICT security rather than purely endeavouring to defend against attacks.

In addition, vulnerability scoring and exploit databases can also be incorporated into the risk assessment strategy for the identification and quantification of vulnerabilities. The Common Vulnerability Scoring System (CVSS), National Vulnerability Database (NVD), Common Vulnerabilities and Exposures (CVE), SecurityFocus Forum, Open Source Vulnerability Database (OSVDB), and Bugtraq Security Database, have been developed to identify and quantify vulnerabilities in a variety of ways with differing focuses. Some methods provide threat warning systems, others provide vulnerability databases, while several vulnerability scoring techniques assist with vulnerability identification.

CVSS has heavily influenced our research and implementation, as the algorithms within the methodology (FIRST.Org, 2019) have been widely incorporated into many vulnerability applications as they have the capacity to assist with assigning numerical values to risks and vulnerabilities. Scores are composed based on three metric groups (base, temporal and environmental). Providing a platform that assigns risk in a standardised manner, including a schema that has the functionality to accommodate industry specifics.

We also incorporate the principals of NVD (Nvd.nist.gov, 2019) which supports automation of vulnerability management and security. NVD is an open repository of vulnerabilities, including essential details in regards to security-related software flaws, security checklists, impact metrics, product names, and misconfigurations. This database is reliant upon the CVE repository; nonetheless NVD expands additional analysis and thus can be considered its superior. While NVD is synchronised to automatically update when new vulnerabilities are identified and published by CVE, it cannot be categorised as a real-time vulnerability and reporting mechanism. As NVD analysts can take as long as two full working days to analyse the vulnerabilities and extend the vulnerability attributes.

Using the metrics within CVSS along with scoring systems and vulnerability databases, in addition to the data security level and energy efficiency grade, we quantify a security grade for each node within the NGI environment based upon the nodes individual software, hardware and firmware. Security is graded on a scale of 0 to 10, with a security grade of 0 being considered the most secure and 10 least secure. Data types retrieved and assigned using its risk probability score to each node include firewall status, intrusion detection system status, encryption status and if used type of encryption, operating system, staff skill level, system update status, anti-virus/security, internet access, data security level etc.

Table 2 demonstrates example parameters and their associated risk probability scores; values are assigned based on the specific domain security requirements and expertise of the security managers and administration. In this example scenario we assigned these constants to reflect our initial network environment, and the values are assigned depending on the importance of the concerned factor and their magnitude.

Table 2. Example parameters and their associated risk probability scores

Risk Type	Risk Probability Score									
	Low Risk		Risk						High	
	1	2	3	4	5	6	7	8	9	10
Firewall Status	True									False
IDS	True									False
Encryption Status	True									False
Encryption Type	AES - 256	TDES - 168			RC2 - 128		WEP - 114			
Operating System	Linux	Mac OS X			Windows Server 2000	Windows 8	Windows XP		HP-UX11i	Solaris
Staff Skill Level	High				Medium					Low
System Updated	True									False
Anti-Virus/Security	True									False
Internet Access	False									True
Data Security Level	1	2	3	4	5	6	7	8	9	10

We consider the data security level as a risk, therefore it contributes to the quantification of the final security grade for the node. These grades are then incorporated into the optimisation process. All parameters used to quantify and assess the appropriateness of the optimised network as shown in Table 3.

Table 3 Simulated optimisation parameters and associated risk probability scores

Centralities	Risk Type and Probability Score	Other
Degree (0-1)	Firewall Status (0 or 10)	Fitness
Betweenness (0-1)	IDS (0 or 10)	Energy Level
Closeness (0-1)	Encryption Status (0 or 10)	Cost
Eigenvector	Encryption Method (0 – 10)	Minimum Path Average
Bridging (0-1)	Operating System (0 – 10)	Security Grade (0 – 10)
	Staff Skill Level (0 – 10)	
	System updated (0 or 10)	
	Anti-Virus/Security (0 or 10)	
	Internet Access (0 or 10)	
	Data Security Level (0 – 10)	
	Identified Vulnerabilities (NVD Score)	

4.3. Robustness function

Robustness analysis and mitigation of network vulnerabilities is very important (Chen et al., 2017, Wu et al., 2018, Jiang et al., 2016, Du et al., 2016, Liu et al., 2016, Liu et al., 2011) to combat cyberattacks in an effective manner. However, existing solutions (Chen et al., 2017, Wu et al., 2018, Jiang et al., 2016, Du et al., 2016, Liu et al., 2016, Liu et al., 2011) may not be as effective as they focus on network connectivity and its features (centralities) only. There are many other important properties of the system, network and their security, which are not considered that they directly affect the performance and security of the entire system. The proposed robustness function is novel and a major part of the proposed solution. It considers network centralities, network parameters and security properties.

The proposed solution can be applied to a network. At the first stage, it takes all properties of the network and models a simulated environment where the robustness function evaluates each node. To determine the optimal secure network, five main criteria are used as a guide. These are the communication security level $S(N)$ outlined in Section 4.1 calculates the security level of a node, highest bridging centrality score $C_R(v^*)$, degree centrality of the network $C_D(G)$, average minimum path length f_{min} , and total cost C . The robustness function is defined as:

$$\phi(i) = [a_1 C_R(v^*) + a_2 C_D(G) + a_3 f_{min} + a_4 C] / S(N) \quad (7)$$

Here v^* is the node with the highest bridging centrality. As the robustness function shows, the main factor is the communications security level achieved. Values for the constants (weights) are as follows:

$$a_1 = 50000, a_2 = 4000, a_3 = 60, a_4 = 10.5$$

The purpose of assigning weights to the robustness function is to give flexibility so it can be adjusted as per application requirement, e.g. a certain application may give preference to data flow compared to local node parameters. In this example scenario we assigned these constants to components that when combined reflect a section of the NGI environment, a_1 represents the highest bridging centrality, a_2 is assigned the centrality degree, a_3 minimum path average, and a_4 associated network cost. As per application, the values assigned to these constants not only depend on the importance of the concerned factor, but also on the magnitude. For example, while centralities generate low numbers, the cost tends to be significantly higher. The lower the robustness, the more appropriate the individual evaluated. It has been ascertained that the robustness increase is inversely proportional to $S(N)$, and that as the other factors increase so does the robustness. The motive being, that we require $S(N)$ to be maximised and all other factors to be minimised. As searching for a lower robustness, means instigating higher communication security, while preserving low cost, degree centrality, bridging centrality, and average minimum path length.

4.4. Optimisation algorithm

4.4.1 Genetic algorithm

The proposed solution uses Genetic Algorithms when measuring the security of overall networks. It helps to reduce the processing time, especially in large scale networks. The basis of the algorithm is to take an initial set of potential solutions, then evolve the set to become a set of best solutions. Through the evolutionary process, inadequate solutions die out, whereas the qualities of the superior solutions are amalgamated and disseminated through new solutions, which are added to the set. Set size remains constant, so as new better solutions are identified, they replace the older inadequate solutions. Random mutation applied to new generated solutions, ensures that the new set of best solutions does not evolve into a set of duplicated solutions. The evolutionary process would continue until a predetermined end criterion is met (Grefenstette, 1986, Kaur, 2016). An outline of the algorithm's pseudo-code is as follows:

Algorithm 1 Pseudo-Code for Genetic Algorithm

```
1: Initialise population with original network (encoded as an individual)  $N_{orig}$ 
2: Next generation array  $N_{Gen}[10]$  equals  $N_{orig}$  plus nine randomly generated populations
3: while stopping criteria is not reached do
4:   for generations  $g$  do
5:     Calculate the robustness of  $N_{Gen}[g]$ 
6:   end for
7:   for generations  $g$  do
8:     if  $F_{best} = 0$  or  $N_{Gen}[g](\text{robustness}) < F_{best}$  then
9:        $F_{best} \leftarrow N_{Gen}[g](\text{robustness})$ 
10:    end if
11:  end for
12:   $N_{Gen}[0] \leftarrow F_{best}$  (next population)
13:  Select three random individuals from previous generation, put in random contest with best individual
  passed to next generation (next population)
14:  Four individuals from new generation are chosen by crossing over two different individuals which have
  been randomly chosen, then passed to next population
15:  Generate new random individuals and add to the new generation until next population equals 10
  individuals
16: end while
17: return best individual from improved solutions
```

The initial population of individuals used by the Genetic Algorithm (GA) is the original NGI environment (encoded as an individual), along with a collection of randomly generated alternatives. For the purpose of our work, the population size is set to be 10 individuals (i.e. the original network plus 9 random networks). Once a population has been generated, every individual is measured by means of the robustness function summarised in Section 3.3.

After evaluating every individual within the population, the best individual is directly passed to the next generation. Three individuals in the new generation are chosen by contest

from the previous generation, the contest passes the best one of these three to the new population. Four individuals in the new generation are chosen by crossing over two different individuals, which have been randomly chosen. Finally, new random individuals are generated and added to the new population, so that the next generation has 10 new populations. After running the cross over and random generation processes, the feasibility of the new individual is checked. Unconnected nodes are prohibited, so if any node is identified as isolated, the new individual is mutated until it is feasible.

New generations are built consecutively. At this point we run the evolutionary process for 2000 rounds, after which we discontinue the application for the GA and the best individual among the remaining solutions is selected.

4.4.2 Ant colony optimisation combined with local search

The ant colony optimisation algorithm is based on the natural foraging behaviour of ants. While the algorithm has assisted greatly when applied as part of the optimisation process, it does have limitations and commonly has to be combined with an alternative optimisation algorithm. The basis of the ant colony optimisation algorithm is to initiate a solution and then update the pheromone trails (i.e. update the comparison parameters). Throughout all iterations, as a new solution is constructed, the pheromone trails are compared (i.e. checking for the optimum path in the graph). After the improved solution is identified the pheromone trail (comparison parameters) is updated with the enhanced parameters. For example, with ants this would be based on the quantity and quality of the food found, trails with a high pheromone would guide ants to the better source. The optimisation process continues until a predetermined end criterion is met (Blum, 2005, Monteiro et al., 2013, Olivas et al., 2017).

The local search method is a simplistic algorithm. The basis of the algorithm is to initiate a solution; the solution is then iteratively evolved, i.e. throughout all iterations the algorithm searches for a better solution, until the predetermined end criterion is met (Monteiro et al., 2013).

The initial population of individuals used by the Ant Colony optimisation combined with Local Search (ANT) is the original NGI environment (encoded as an individual), along with a collection of randomly generated alternatives. Similarly, we generate and compare 10 individuals for each cycle of the evolutionary process. Once the population has been generated, the solution trail is assigned the original networks comparison parameters (i.e. this is the best solution we begin with hence these are the parameters that need to be compared and improved). Every individual is then measured by means of the robustness function (described in Section 3.3).

After evaluating every individual within the population, each solution is compared against the best robustness, in an attempt to find an improved generation. Should the cycle produce a better solution, the solution trail is then updated with the new solutions comparison parameters. After each cycle, we compare each improved generation's parameters in the solution trail, placing them into descending order, ensuring that we only

keep the 5 most improved solutions. New generations are built consecutively, and the process runs for 2000 rounds. We then discontinue the application for the algorithm and the best individual among the remaining solutions is selected, along with reporting the 5 most improved solutions and identify their respective costs.

An outline of the algorithm's pseudo-code based on a combination of ant colony optimisation and local search is as follows:

Algorithm 2 Pseudo-Code for Ant Colony Optimisation combined with Local Search

```
1: Initialise population with original network (encoded as an individual)
2: Calculate original populations robustness  $F_{old}$ 
3: Initialise parameters
4: Initialise solution trails
5: while stopping criteria is not reached do
6:   Generate a new random solution
7:   Calculate new solutions Robustness  $F_{new}$ 
8:   Calculate parameters
9:   if  $F_{new} < F_{old}$  then
10:     $F_{old} \leftarrow F_{new}$ 
11:    Update solution trails with parameters
12:   end if
13: Compare all solutions sort into descending order
14: end while
15: return five improved solutions and identify solutions with their respective costs
```

4.4.3 Tabu search

Tabu search is a metaheuristic search method, which uses local search methods for optimisation, along with adaptive memory to explore beyond local optimality and to generate dynamic search method performance. The basis of the search is to prevent the method from re-examining solutions that have already been considered, and to ensure that inadequate solutions are not developed further. Parameters of preference can also be introduced, influencing the search into producing a more favourable solution. Tabus tend to only be stored as a limited quantity, as typically there are several possibilities and tabu lists can quickly grow in size, making storage of these parameters and comparison expensive. Therefore, restricting the tabu list to only recent improvements and preventing reverse evolution ensures quick and non-costly optimisation. The optimisation process continues until a predetermined end criterion is met (Brownlee, 2011, Monterio et al, 2013).

Initial population used by the Tabu optimisation process (TABU) is the initial NGI environment (individually encoded), along with nine randomly generated alternatives. Once the population has been generated, the tabu list is assigned our predefined comparison parameters from the original environment, as at this stage this is the best solution and we aim to prevent inferior solutions from being considered. Each solutions predefined parameters are then compared against the tabu list, if parameters match the tabu list they are

dropped. Else, if parameters are not tabu, then we calculate the robustness of the solution by means of the robustness function (Section 3.3).

We then compare the solutions robustness against the best robustness, to ensure that the generation is improved. Should the cycle produce a better solution, then the robustness of the new solution replaces the best solutions robustness, and at the end of the cycle the tabu list is updated ensuring that only improved solutions are considered. New generations are built consecutively, and we run the search for 2000 cycles. The search application is then discontinued, and the best individual among any remaining solutions is presented. An outline of the search method's pseudo-code is as follows:

Algorithm 3 Pseudo-Code for Tabu Search

```

1: Initialise population with original network (encoded as an individual)
2: Calculate original populations robustness  $F_{old}$ .
3: Initialise parameters  $P_{best}$ 
4: Generate tabu list  $\leftarrow P_{best}$ 
5: while stopping criteria is not reached do
6:   for generations  $g$  do
7:     Let  $g$  construct new random solution
8:     Calculate parameters  $P_{new}$ 
9:     if  $P_{new}$  not tabu then
10:      Calculate new solutions Robustness  $F_{new}$ 
11:      if  $F_{new} < F_{old}$  then
12:         $F_{old} \leftarrow F_{new}$ 
13:      end if
14:    end if
15:    Update tabu list
16:  end for
17: end while
18: return best solution  $P_{best}$ 

```

5. Applying the methods: simulation results

Optimisation algorithms have the ability to manage large, complex optimisation problems, with the focus of our work being that of the NGI specifically smart cities. The optimisation algorithms, principal concepts and robustness function described in Section 3 have been implemented as part of the proposed solution. This tool was developed for proof of concept and to critically analyse the effectiveness of the techniques discussed in this paper. This work has already been applied to a real network, where agents were installed on the local machine and connected to the server where the SCRAM application operates. It considers connectivity, collaborative analysis reports and warning systems, securing globalised network view, accessibility between collaborative organisations, congestion avoidance and control, and limiting the impact of resources used for processing.

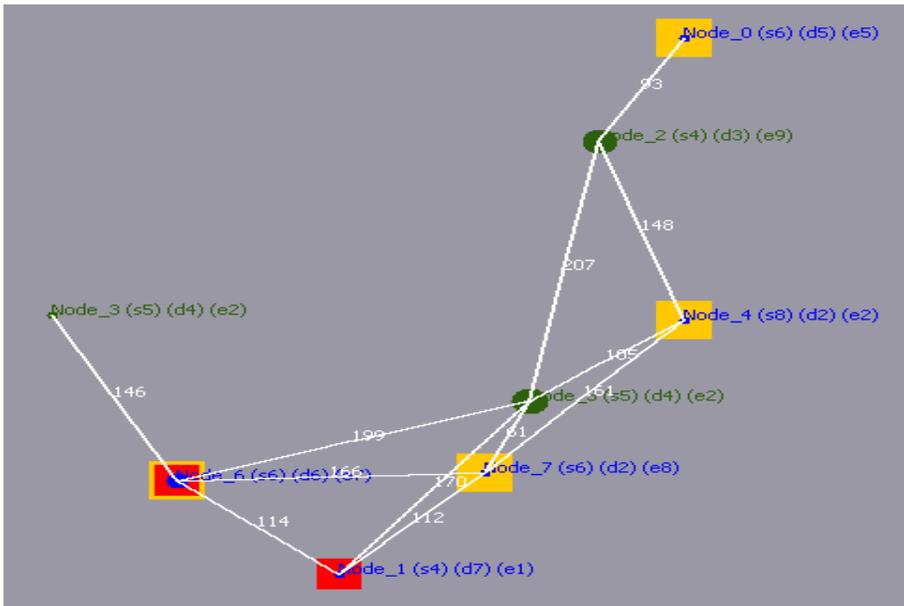


Fig. 4-a. Primary simulated NGI environment, Network A Security graph

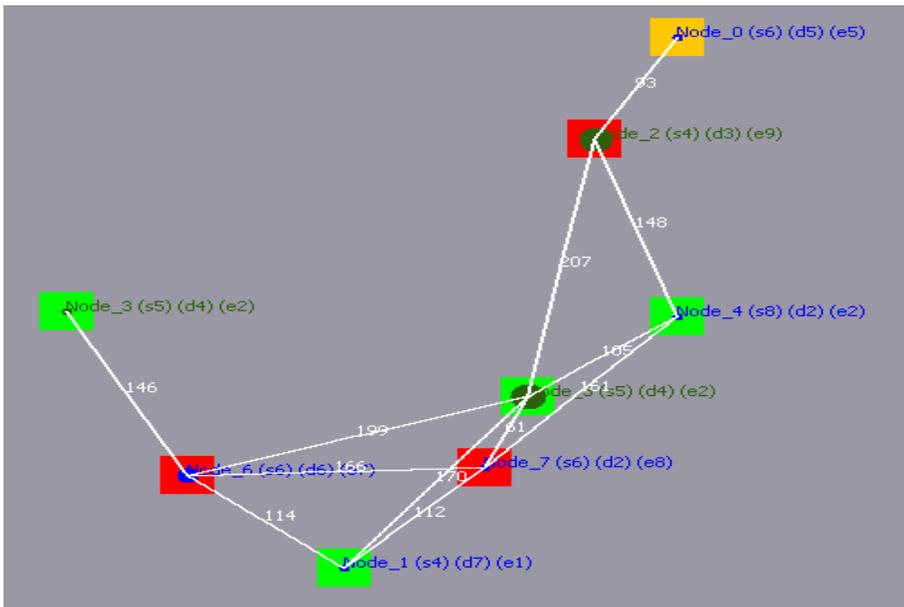


Fig. 4-b. Primary simulated NGI environment, Network B Energy Efficiency graph

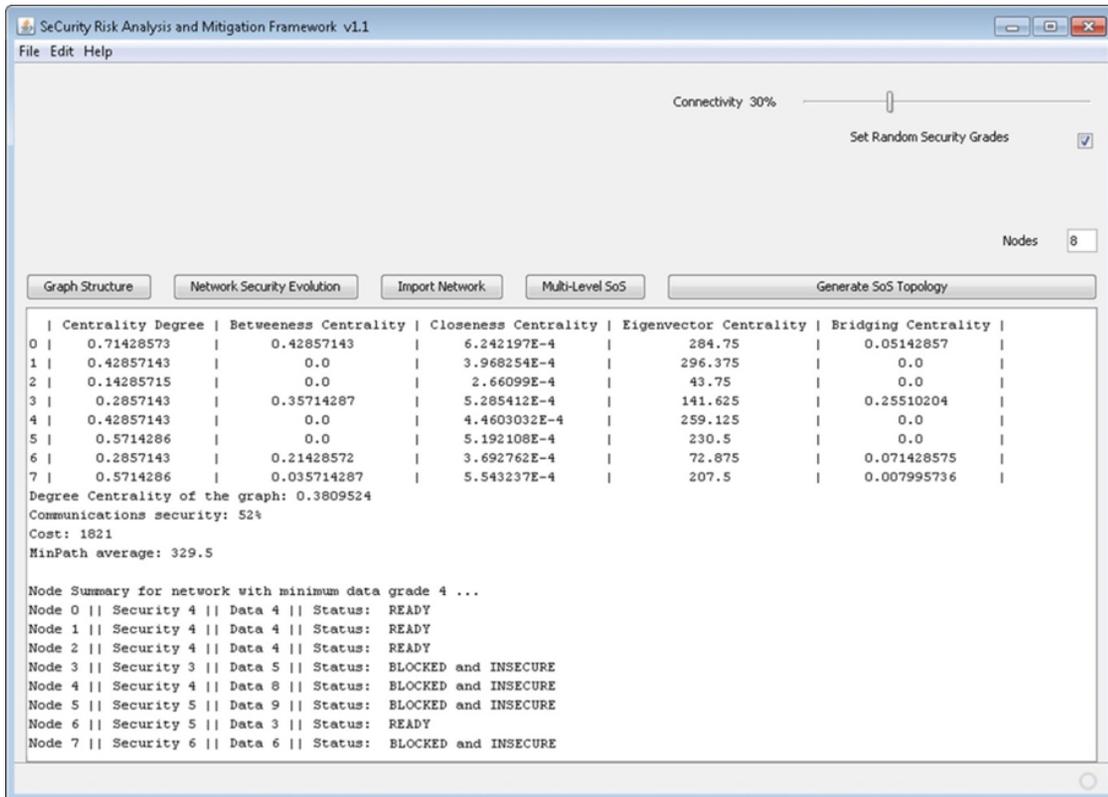


Fig. 4-c. Primary simulated NGI environment data report

Similar to the work of Ali et al. (2016), Rullo et al. (2017), Yan et al. (2017) and Yao, et al. (2017), we use simulation to generate our initial results and conduct evaluation, which ensures we do not negatively impede the functionality of a deployed network, while evaluating the framework and implemented algorithms capabilities. The framework also provides an inexpensive simulation model to conduct experiments within, allowing us to study the behaviour of the systems and techniques.

Figure 4-a visualises the primary network (Network A), displaying key parameters so we can examine the graph intuitively, this figure depicts the security of the network. Figure 4-b visualises the same simulated NGI section, however, exhibits the energy levels of the nodes. Figure 4-c shows part of the data report which is quantified and helps form the networks that are visualised in Figure 4-a and 4-c. Table 4 defines the visualised parameters used to generate the undirected graphs.

The simulated network consists of 8 static nodes with a low connectivity of 30%, and is formed using a variety of ICT devices which include sensors and mobile devices. This section represents IoT devices within a smart city, with each device randomly assigned the relevant node software, hardware, and firmware parameters such as the type of operating system, energy level, data access grade, whether it supports encryption, Internet access, incorporates firewalls, IDS, and anti-virus/security, and if the node has been completely updated or has vulnerabilities.

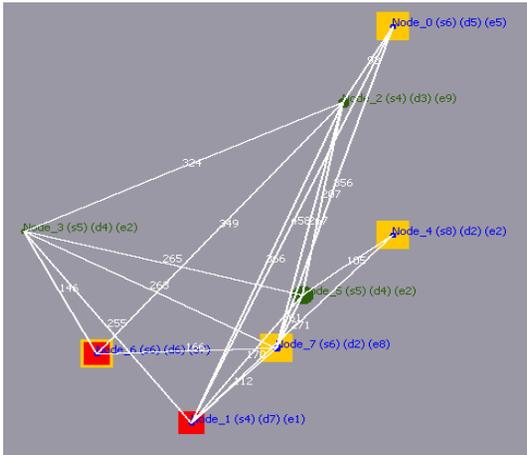
Then the framework randomly assigns all nodes with a security level and connects them via a series of primary links. It then quantifies the networks degree, betweenness, closeness, eigenvector, and bridging centralities, the communications security, minimum path average, and the networks associated cost. Our framework, then assigns a random network data level, which all nodes will be compared against replicating data access control principals.

For security grades/levels to be accurate, it is vital that we identify vulnerabilities that have the potential to expose nodes to risks, which in turn can negatively impede the entire networked topology. Vulnerabilities often are identified using a vulnerability scanner, allowing for vulnerability scoring and exploit databases to be incorporated into the risk assessment methodology. Conducting risk assessment in an NGI environment is highly problematic, great consideration must be taken when applying methods directly to systems which are deployed or deemed critical, as methods could impact the collaborative components and their ability to interoperate.

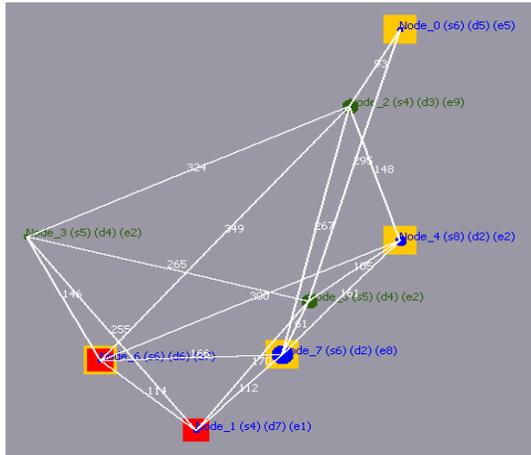
We incorporated risk assessment into SCRAM to simulate vulnerability identification, and assign reported NVD vulnerabilities to nodes, in a random method based on the type of device and its operating system. Simulating this scan means security scores are quantified with greater accuracy, SCRAM then generates detailed reports on all security parameters, centralities and identified vulnerabilities with their associated CVSS v3 base scores.

Table 4. Security Risk Analysis and Mitigation (SCRAM) frameworks visualised parameters

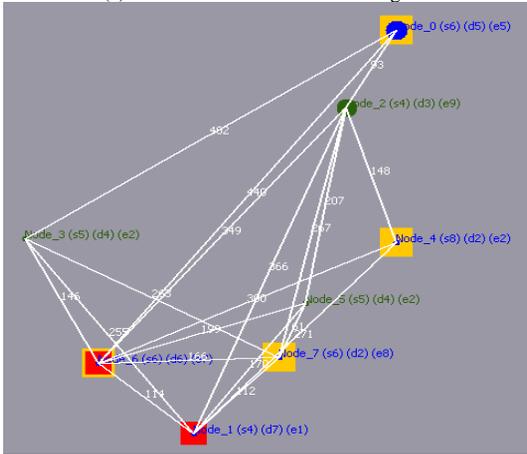
Graph	Parameter	Symbol	Description
All graphs	Scanned node no vulnerabilities		Dark green node/tag
	Scanned nodes identified vulnerabilities		Blue node/tag
	Unscanned node		Dark red node/tag
	Node size represents quantified bridging centrality, i.e. small nodes low and large nodes equal high.		
Security	Insecure node		Node encased with a solid orange box
	Blocked node		Node encased with a solid red box
	Blocked and insecure node		Node encased with a solid red box with orange border
Energy efficiency	High node energy level		Node encased with a solid green box
	Medium node energy level		Node encased with a solid orange box
	Low node energy level		Node encased with a solid red box



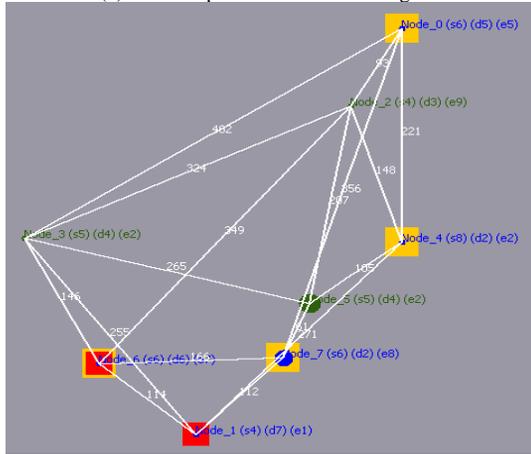
(a) First mutated candidate using GA



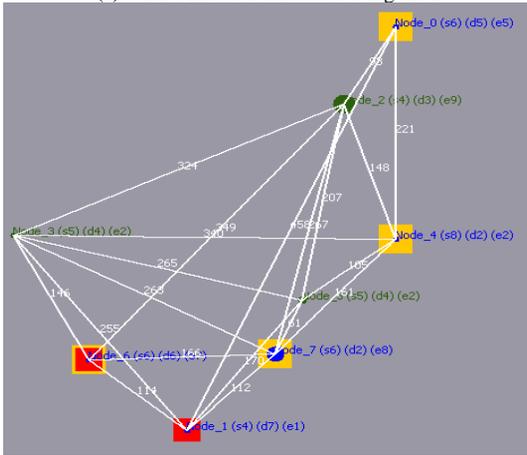
(b) Final optimum candidate using GA



(c) First mutated candidate using ANT



(d) Final optimum candidate using ANT

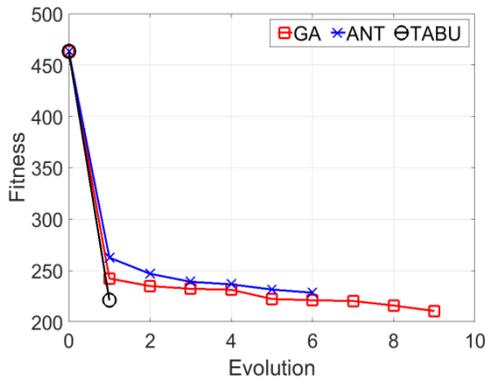


(e) First and Final optimum candidate using TABU

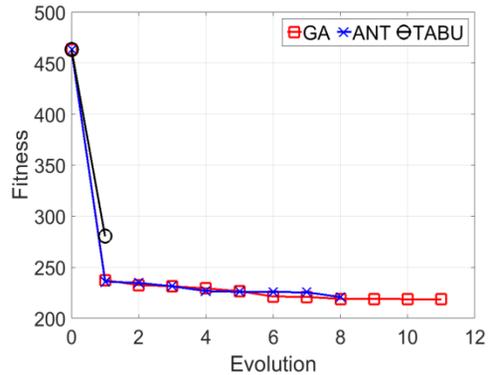
Fig. 5. Security optimisation evolutions for Network A visualising all improved solutions, prioritising security and data access control

When each algorithm is applied it is integrated with the methods robustness function, then the network is evolved into a set of best solutions as described in Section 4. Through the evolutionary process random mutations are made to each generated solution, and these configurations are produced from a single run generating 20,000 evolvments. Figures 4 and 5 visualise the first and the final optimum solution for each of the applied algorithms, as examples of how the network has evolved in comparison to the original network.

5.1. Network robustness



(a) Robustness monitor for Network A optimisation focusing on security and data access.



(b) Robustness monitor for Network B optimisation focusing on energy consumption.

Fig. 6. Robustness monitor graphs for the applied algorithms

Throughout the optimisation process each node is measured by means of the robustness function (Section 3.3), Emphasis is placed on the robustness level of the network as it assists the algorithms to produce the next generation of improved solutions, utilising the key parameters of individuals being selected. Other factors are also reported and considered such as the degree centrality of the graph and energy efficiency, and key parameters are also reported and analysed as standalone risks, as illustrated in Section 4.3. As the robustness level is a combination of parameters, it provides an intuitive overview of the networks security suitability and risks posed, and a demonstrative measure of general improvement. Therefore, low robustness scores show evolved improvements so are considered, while high robustness scores demonstrate degeneration so are omitted. There is no guarantee that as a network is mutated improvements to the network and its robustness will be achieved, even when evolvment is positive it can take a vast number of cycles before progress is attained.

The robustness graphs in Figure 6 visualise network robustness when each of the algorithms was applied to both scenarios throughout all evolutionary processes. These graphs record a notable reduction in network robustness, for both Networks A and B. When the algorithms were applied they randomly mutated new candidates in a positive method, meaning the reported improved solutions are more appropriate. The robustness monitor for Network A (Figure 4-a) shows the original network had a robustness score quantified as

463.3917. The GA achieved a 56.51% improvement; ANT achieved a 50.72% improvement, while TABU improved robustness by 52.31%.

The robustness for Network B (Figure 4-b) was also quantified as 463.3917, in some simulations, we see marginal fluctuations of difference between the original robustness scores because the framework is quantifying the robustness focusing on different key parameters. GA improved the robustness by 52.85%, ANT improved robustness by 52.37%, while TABU reduced robustness by 39.52%. In both scenarios robustness is improved from the first reported evolution for GA and ANT, ranging between 39.52% and 52.31%. This positive development continues to advance throughout the optimisation process.

5.2. Data analysis

During evolution stages the applied principals search for an optimal combination, using processes that removes and replaces links within the WSN. Figures 5-a, 5-c, 5-e, 6-a, 6-c and 6-e, visualise the first improved generations which assure communication security, each showing an increase of communication links. The cost increase (Figures 8-a and 9-a) for both scenarios reflects this growth of communication paths, with the applied algorithms increasing the cost of Network A on average by 104.8% and Network B by an average of 104.2%. It is essential that the optimisation process when adding and removing links, balance connectivity with improvements to the WSN robustness and security, while unduly impacting centrality factors. The framework is not attempting to revise cost, simply associate network cost with suggested WSN modifications. Network A which prioritises security and data access, shows that GA has the lowest costing optimal solution (Figure 7-a) increasing by only 98.04%. Network B which prioritises energy levels, shows that ANT has the lowest costing optimal solution (Figure 9-a) resulting in an increase of 88.59%.

Through the improved robustness, the algorithms and processes sustain low degree centrality (Figures 8-b and 9-b) for both scenarios. While the networks optimal solutions do not have the lowest degree centrality score, each solution with the exception of TABU, has a reported improved centrality score compared to the original network. For example, Network A (Figure 8-b) both GA and ANT optimum candidates decrease degree centrality by 62%. While degree centrality is not a key parameter used to quantify network robustness, as the algorithms process network mutation, they reject mutated candidates that critically increase degree centrality, i.e. minor negative increases are acceptable and considered to be within a tolerable range.

There are notable fluctuations between reported candidates for minimum path average (Figures 8-c and 9-c). In both scenarios the only increase in minimum path average occurred when TABU was applied to Network B which focuses on energy levels. This negative increase is reflected in the TABUs robustness score (Figure 7-b) which is slightly higher compared to the other algorithms robustness scores. Minimal path average reduced by 24.25% using GA and 25.43% using ANT on Network A, and by 28% using GA and by

15.29% using ANT on Network B. These scores directly correlate to the new established links between nodes.

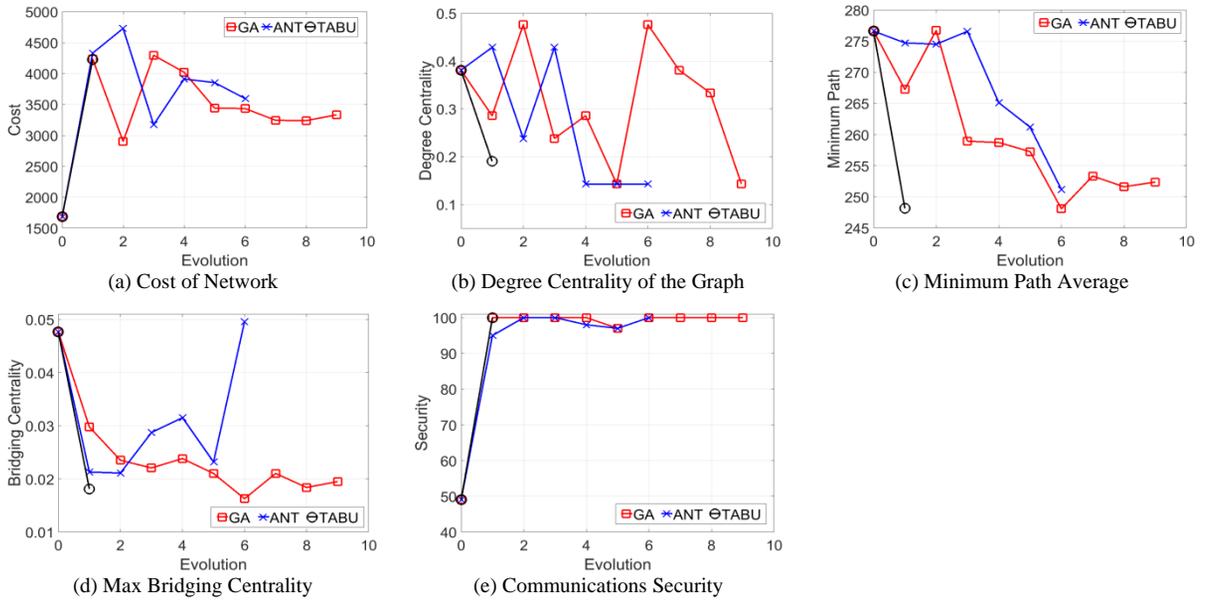


Fig. 8. Network evolution results comparison for Network A data

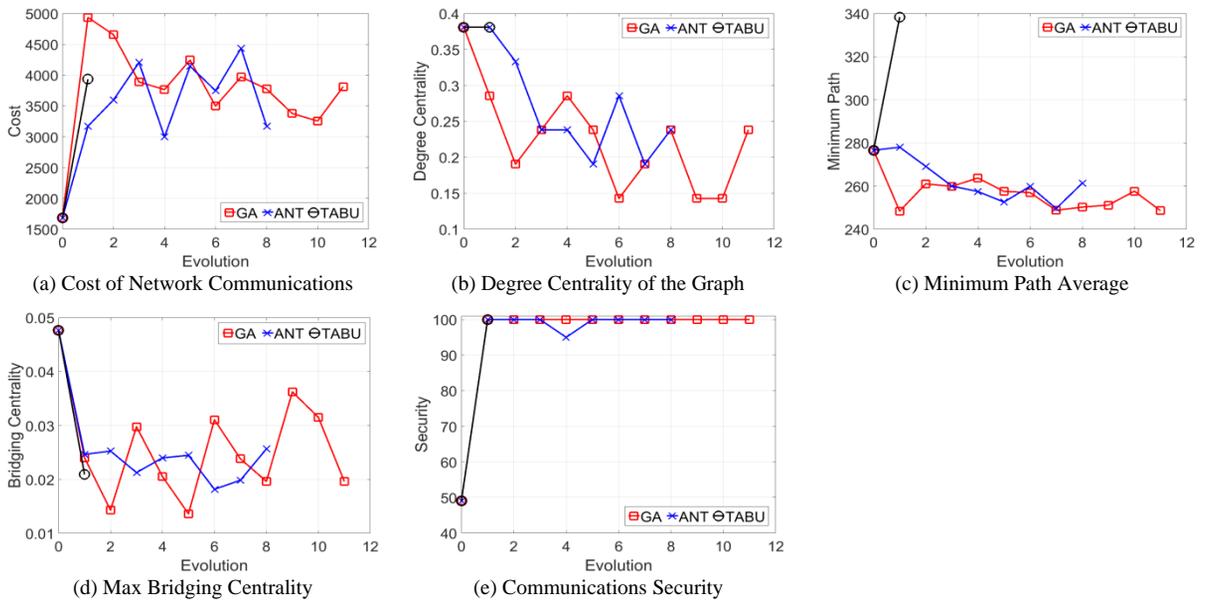


Fig. 9. Network evolution results comparison for Network B energy

Analysing bridging centrality (Figures 8-d and 9-d) there are significant fluctuations between candidate scores for both scenarios and all algorithms. Network A (Figure 8-d) indicates that the final optimum solution when ANT was utilised has a minor increase of 4.17% in comparison to the original network. In contrast to GA, which decreased bridging centrality by 59.09% and TABU which decreased by 62.03%. Analysing Network B (Figure 9-d) each of the applied algorithms generated final solutions with decreased bridging centrality scores, GA decreased by 58.79%, TABU decreased by 56.1%, while ANT had the lowest decrease of 46.15%. Despite the single minor increase, which is within a tolerable range, the analysis corroborates that as the WSN is mutated and algorithms applied, each of the methods support the mutation of the network ensuring that evolvments that negatively impede developments are rejected. As evident by not only sustained low centralities, but also in the improvement to the robustness score.

Table 4. Aggregated node centrality scores for all improved evolutions of Network A

Evolution	No. Links	Cost	Robustness	Degree	Betweenness	Closeness	Eigenvector	Bridging
0 (Primary Network)	12	1682	463.391	0.428	0.102	0.00054	190.984	0.0165
GA Evolution 2 low-cost	14	2901	234.834	0.5	0.066	0.00053	374.265	0.0063
GA Evolution 9 optimum	17	3331	210.548	0.607	0.049	0.00058	465.484	0.0083
ANT Evolution 3 low-cost	16	3172	239.015	0.535	0.062	0.00053	401.484	0.0073
ANT Evolution 6 optimum	17	3595	228.368	0.607	0.058	0.00058	488.640	0.0145
TABU Evolution 1 optimum	20	4225	220.986	0.714	0.035	0.00059	686.593	0.0051

Table 5. Aggregated node centrality scores for all improved evolutions of Network B

Evolution	No. Links	Cost	Robustness	Degree	Betweenness	Closeness	Eigenvector	Bridging
0 (Primary Network)	12	1682	463.391	0.428	0.102	0.000545	190.984	0.0165
GA Evolution 10 low-cost	18	3255	218.686	0.607	0.058	0.000572	462.75	0.0122
GA Evolution 11 optimum	20	3813	218.479	0.678	0.044	0.000592	609.015	0.0058
ANT Evolution 4 low-cost	16	2998	226.595	0.535	0.058	0.000568	376.906	0.0119
ANT Evolution 8 optimum	16	3172	220.717	0.535	0.066	0.00056	433.281	0.0088
TABU Evolution 1 optimum	16	3935	280.238	0.571	0.058	0.000314	544.109	0.0092

We observe for both scenarios, there are significant increases to communication security from the first evolved candidate (Figures 8-e and 9-e), with minor fluctuations occurring from 95% to 100% for both networks. Each of the optimum generated solutions report 100% secure network communications, increasing security by 51%.

Alternatively, via the use of the frameworks detailed reports we can evaluate each node and analyse how individual centralities are impacted due to network evolution, Tables 4 (Network A) and 5 (Network B) present aggregated node centrality scores for the primary, lowest costing, and optimum network. Evaluating individual nodes assists in determining how individual nodes are impacted compared to analysing the network as a single entity. An example report is shown in Table 6, reporting data for Network A when GA was applied, showing individual node bridging centrality for each improved evolution.

We ascertain that bridging centrality utilising GA in both scenarios decreased by over 58%. However, for Network A average node bridging centrality only improved by 49.38%, while Network B improved by 64.87%. In both scenarios utilising ANT, Network A decreased bridging centrality by 63% yet the average node centrality only improved by 12.23%, while Network B reported an 56.1% centrality improvement while its average node bridging centrality scored an 46.5% improvement. The report indicates that in all instances the values are in an acceptable range, but these reports provide a solution for quick analysis to assist with decision making processes.

Additionally, these reports help ascertain the values of the most optimum solution, and can identify if there are cheaper alternative candidates that are more cost effective to implement and don't impact centrality values and security, identifying suitable alternatives than optimum solutions. Reviewing TABU results, there are no cheaper alternatives to consider, due to algorithms rigid methods failing to yield alternative optimised solutions. When GA and ANT is applied to both scenarios, cheaper candidates to implement are reported, which improve network robustness and security compared to the original Smart City WSN.

However, just because cheaper alternatives are established, they should only be considered if they maintain a series of alternative links between secure nodes, thus results are compared against the undirected graphs.

Table 6. Security optimisation node bridging centrality scores for Network A evolutions when GA is applied

	node 0	node 1	node 2	node 3	node 4	node 5	node 6	node 7
Evolution0	0	0	0.046	0	0	0.047	0.030	0.007
Evolution1	0	0	0.017	0	0	0.029	0	0.012
Evolution2	0	0	0.008	0	0	0.023	0	0.019
Evolution3	0	0	0.014	0	0	0.022	0	0.013
Evolution4	0	0.011	0.023	0	0.0088	0	0.020	0.010
Evolution5	0	0.019	0.015	0	0.0064	0.010	0.006	0.021
Evolution6	0	0	0.015	0	0	0.016	0	0.006
Evolution7	0	0.009	0.013	0	0.0210	0.019	0	0.019
Evolution8	0	0	0.010	0	0	0.014	0.010	0.018
Evolution9	0	0	0.015	0	0.0111	0.014	0.006	0.019

5.3. Observations

Analysing the undirected graphs that focus on security and data access (Figure 5), we intuitively identify in Figure 5-a, that the first evolved candidate produced using GA increased the number of links between nodes from 12 to 17, ensuring that a secure route was established between all secure nodes. Figure 5-b shows that as the WSN evolved further, a secure route between nodes 2, 3, and 5 was maintained using 17 links. Analysing ANT we identify that the first candidate increased the number of networked links (Figure 5-c) from 12 to 19, and the optimum solution (Figure 5-d) maintained a secure route between nodes and is formed using 17 links. Figure 5-e visualises the only candidate produced using TABU, this algorithm establishes a secure route between nodes using 20 links, which is greater than solutions generated via GA and ANT. Reviewing the undirected graphs that focus on energy efficiency (Figure 6) we see similar characteristics.

For each final optimum solution for Network A (Figures 5-b, 5-d, 5-e) we intuitively see that all candidates have multiple links between secure nodes, meaning if a secure link was removed, a single secure route will be maintained. Limiting the risk of a single point of failure, and ensuring that nodes are unlikely to become isolated and cut off from the remainder of the WSN. Should multiple secure links be removed, there are alternative paths between secure nodes. However, data will have to traverse via nodes which have been quantified as insecure placing the data at risk. Fortunately, these links have been identified and reported via the method, and visualised in the undirected graph, providing advanced warning and an opportunity to make changes to improve the security of these nodes.

Likewise, final optimum candidates for Network B (Figures 6-b, 6-d,) identify significant links maintained between high energy nodes. In Figure 6-d there is only a single path between secure nodes. Should a single link be removed, then there are no secure paths for data to traverse, and data will be transmitted across paths between insecure nodes.

For Network B the priority of the principles and algorithms was to quantify and optimise the WSN based on node energy efficiency, as well as to maintain low centralities, high network security, data access violations, and node vulnerability. While this has been achieved, due to the methods prioritisation of energy efficiency there is a lack of alternative paths between secure nodes that are present within optimum candidates of Network A. Which is expected as the methods priority is shifted from network vulnerabilities and data access. Figure 6-b is the only exception, the optimum solution utilising GA shows there are multiple links between nodes 2, 3 and 5, therefore if a single link was removed nodes can maintain a secure path for data to be routed. The applied algorithms and principles adequately can also support network optimisation based on energy efficiency and can succeed in extending network life, evident from our initial simulation results.

In WSN while the data access control problem would be less likely to be a priority over energy efficiency, we aim to improve data flow security. Implementing the new methods to focus on energy efficiency we see unstructured behaviour forming for both GA and ANT. This is due to the optimisation process focusing on the energy efficiency levels and

combining security and data access grades into the algorithms process. As random mutations occur while the algorithms are prioritising node energy levels, ensuring that high energy nodes stay linked in case low level nodes fail, the algorithms still have to ensure as mutations are made to the network, security and data access control is maintained.

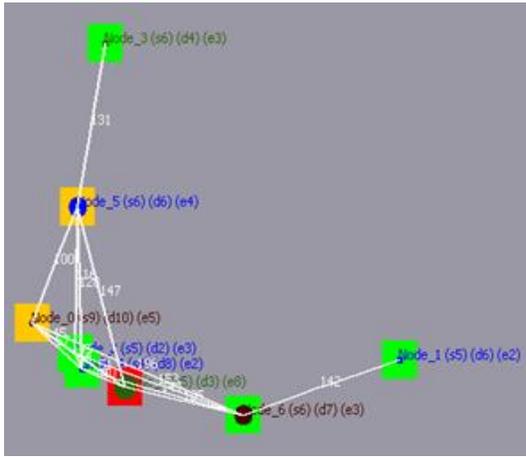
While TABU ensures a quick and non-costly optimisation process, completing its run in 38 seconds compared to GA which completed in 1 minute and 4 seconds and ANT that completed in 45 seconds, it fails to report or consider inadequate solutions, and only improved solutions are developed further. This is due to its restricted comparison parameters that must be matched or improved. The tabu list influences cycles preventing reverse evolution from being considered in order to improve processing time and costs, but as we analyse results we note that other configurations can be appropriate.

Should organisations have financial restrictions in regards to network security, because the framework did not only just present the optimum solution but alternative candidates utilising GA and ANT. These alternative evolutions can be considered for adoption, in the awareness that the framework has optimised and improved the overall robustness of the network. These evolutions and recommended improvements assure network security and reduce potential risks to data communications.

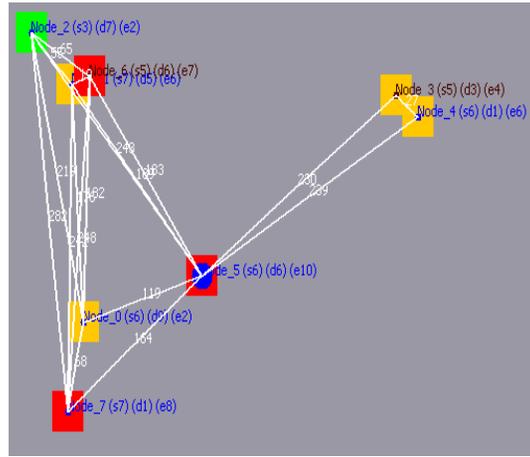
While new communication links help to establish secure routes across the WSN, as well as supporting node connectivity, they negatively impact network security as they are the basis for additional risk factors. In addition, these new communication links come at a price, as in order to achieve improved network robustness and lower centralities, there is a significant increase in network communication costs.

5.4. Simulation analysis

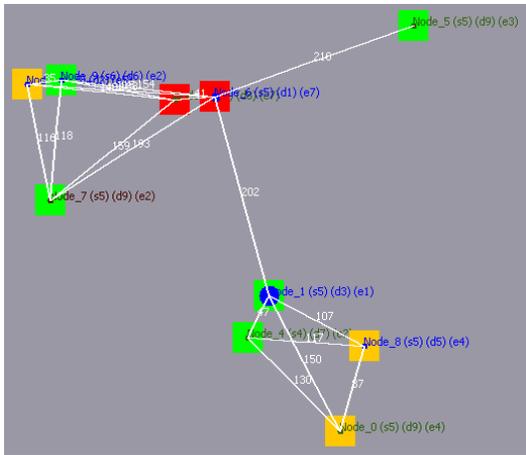
We have run six different simulations which reflect sections of an NGI environment, each of which is based on a WSN or IoT topology. Figure 10 visualises these six sections in a series of undirected graphs, which we have experimentally tested by applying the principals and algorithms discussed in Sections 2 and 3. These graphs visualise all implemented and tested network node energy efficiency levels, which not only observes the data access control problem, security levels and identified vulnerabilities. But also focuses on mutating the network during the process considering each nodes energy efficiency level, in an effort to extend the life of the network.



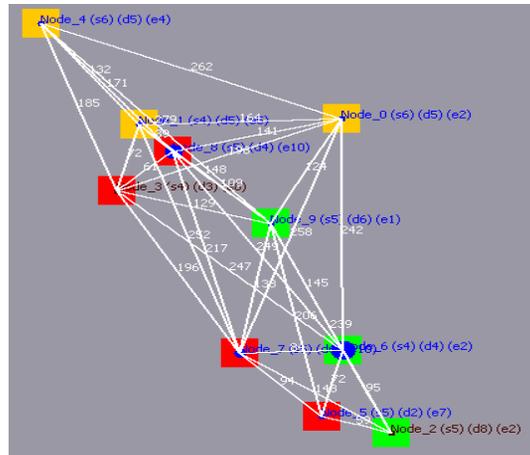
(a) 8 nodes 30% connectivity



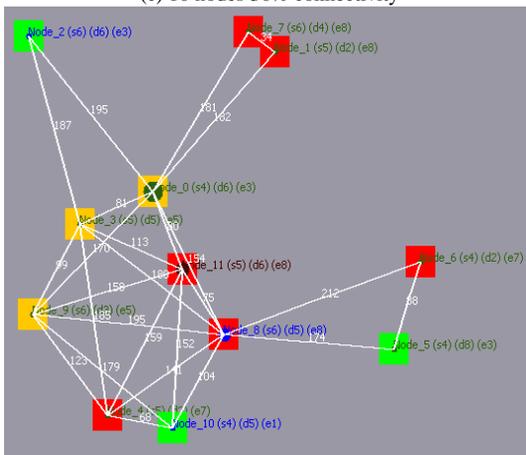
(b) 8 nodes 40% connectivity



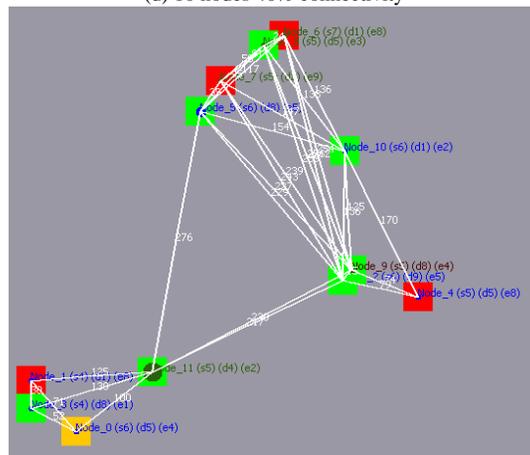
(c) 10 nodes 30% connectivity



(d) 10 nodes 40% connectivity



(e) 12 nodes 30% connectivity



(f) 12 nodes 40% connectivity

Fig. 9. Simulated smart city networks used in experimental visualising node energy efficiency levels, and their final optimal evolution

Each NGI section contains 8, 10 or 12 nodes, with a low connectivity level of either 30% or 40%. Nodes are then assigned their relevant parameters, data access level, security grade, energy efficiency level, and are connected via a series of primary links. The NGI sections were then imported back into SCRAM and we applied GA and ANT to each scenario consecutively.

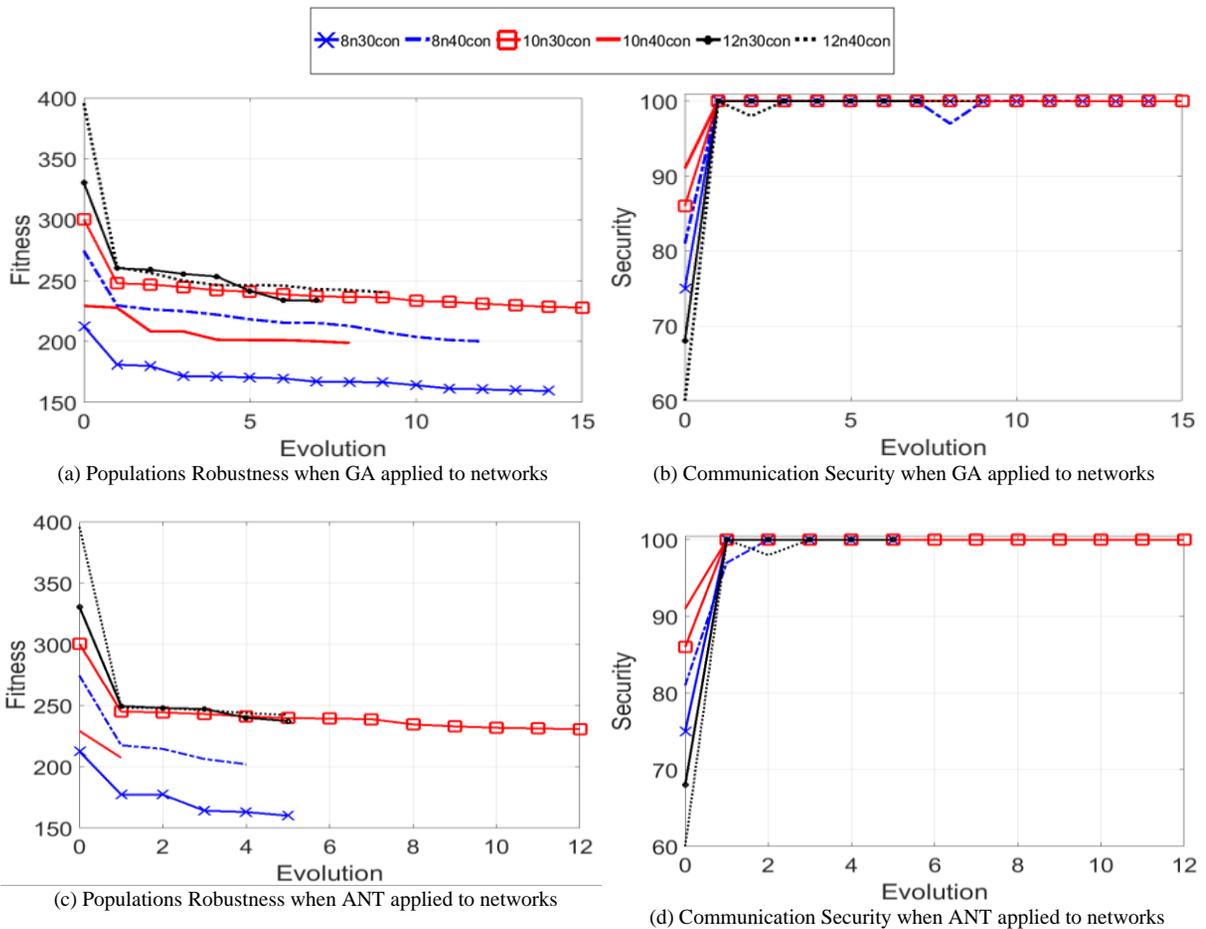


Fig. 11. Network evolution results for experimental network simulations

For these investigations, we did not utilise TABU as we have ascertained it does not yield adequate results or report alternative optimised candidates. In each instance, we prioritised energy efficiency as part of the optimisation process, after initial simulation results showed a great capacity for optimisation, and in an attempt to extend the network life in NGI scenarios. Figures 11-a (GA) and 11-c (ANT) visualise each of the networks population robustness during the entire evolutionary process. These graphs clearly indicate

a notable reduction on the network robustness for all scenarios, corroborating that all final optimal solutions are more appropriate as their robustness grades have been quantified lower. Similar to the above discussed results, when we analyse the evolution results in Table 7 and Figure 11 we ascertain that GA produced more evolved candidates for analysis, and for all six scenarios GA generated mutated optimum solutions with lower robustness scores in contrast to ANT.

For example, when GA was applied to Network A (Figure 10-a) the network robustness improved by 24.97%, which is 0.4% more than ANT, and when GA was applied to Network D (Figure 10-d) the network robustness improved by 13.28% which is 4.13% greater than ANT. On average GA had a 1.4% better optimal robustness score for scenarios in comparison to ANT. Each of these mutated optimal solutions not only increases the robustness of each scenarios topology, but also increases network communication security as visualised in Figures 11-b and 11-c. It is evident that after the first reported candidate security never drops below 97%, and only 4 of the evolved candidates report a security score that does not equal 100% evident in Figures 11-b and 11-d.

While the replacement and removal of communication links balances connectivity with advances to security and robustness, these improvements impact the overall cost of the communication network (Table 7). In some instances, we note that evolution can decrease or cause minimal cost increases, e.g. Network D (Figure 10-d) both GA and ANT reduce network cost. Similarly, through the analysis of the reported evolutions for each network, there were alternative cheaper reported evolved candidates.

Analysing the degree centrality for the simulated NGI environments, we ascertain that the applied algorithms during the optimisation process have mutated the networks and selected only evolutions that lower and maintain low degree centrality with the exception of Network F (Figure 10-f) when GA was applied, i.e. minor increase from 0.272727 to 0.290909, which is a 6.67% increase.

Minimum path length for each of the optimum solutions reported in Table 7; demonstrates that the applied processes have assisted in evolving each of the networks and ensured that only candidates that improve the network, or maintain centralities that are considered with an acceptable range are selected as suitable reported candidates. In all but three networks, minimum path average is reduced.

The mutation of the communication links within each scenario greatly influences bridging centrality, and throughout the evolution for each network, we noted fluctuations of bridging centrality scores. This is expected due to the removal and replacement of communication. In all instances with the exception of Network B (Figure 10-b) when GA was applied (20% increase) and Network D (Figure 10-d) when both algorithms were utilised (GA 27.06 %, ANT 56.79% increase), we see a decrease in bridging centrality for all optimal evolutions. The applied algorithms and processes when establishing secure communication links between nodes are influenced by the security score of the node and data access control. The mutated networks reflect the decisions of the applied algorithms

and processes, along with the positions of the nodes within each of the network topologies and the communication links which nodes are reliant upon for data transfer.

The results for these simulations are similar to the reported case study outlined in Section 4. These simulations and generated reports provide sufficient data and initiate warnings, so minor fluctuations and increases are thoroughly reported and identified to assist with all decision making processes. Due to network mutations we cannot guarantee that evolvments will not negatively impact centrality scores, what is evident is that the algorithms and processes are ensuring that only acceptable negative centralities are considered as part of the wider evolvment process and robustness evaluation.

As optimal evolutions for each NGI network maintain a series of prime links between nodes that have good energy efficiency, this ensures that data communication transfer can be conducted via nodes that have high energy efficiency and bypass low energy nodes. This means lower energy nodes will not be responsible for transferring or processing unnecessary volumes of data, and will extend the life of these nodes and the NGI environment in which they play a key role.

Table 7. Network evolution results comparison for experimental NGI simulations

Evolution	Cost	Robustness	Degree	Min Path Average	Bridging	Security
8 node 30% connectivity (Network A)	1664	212.583	0.190	192.821	0.028	75
GA Evolution 14 optimum	2801	159.508	0.095	181.714	0.019	100
ANT Evolution 5 optimum	2745	160.142	0.095	181.928	0.022	100
8 node 40% connectivity (Network B)	2950	274.558	0.476	253.25	0.025	81
GA Evolution 12 optimum	2899	200.206	0.047	241.642	0.030	100
ANT Evolution 4 optimum	3293	202.170	0.285	231.5	0.017	100
10 node 30% connectivity (Network C)	2317	300.471	0.333	296.6	0.085	86
GA Evolution 15 optimum	5416	227.940	0.111	264.777	0.015	100
ANT Evolution 12 optimum	3628	230.737	0.305	269.977	0.037	100
10 node 40% connectivity (Network D)	5153	229.322	0.361	219.977	0.016	91
GA Evolution 8 optimum	3869	198.870	0.194	233.622	0.020	100
ANT Evolution 1 optimum	4758	207.437	0.138	231.977	0.025	100
12 node 30% connectivity (Network E)	3669	330.591	0.4	275.106	0.023	68
GA Evolution 7 optimum	6980	233.850	0.163	263.939	0.015	100
ANT Evolution 5 optimum	5939	237.328	0.327	272.394	0.017	100
12 node 40% connectivity (Network F)	4783	395.788	0.272	271.697	0.043	60
GA Evolution 14 optimum	6763	240.481	0.290	270.712	0.014	100
ANT Evolution optimum	6113	242.278	0.236	283.181	0.0189	100

6. Conclusion & Future Work

As the complex systems or networks are dynamic and adaptive with emerging behaviours therefore it is very likely that it will produce a large number of vulnerabilities. Therefore, any solution proposed to counter these vulnerabilities must be dynamic. The proposed solution works in a dynamic fashion using evolutionary algorithms and probabilistic techniques and optimises the level of security in NGI environments and extends network life, while considering factors such as energy efficiency, access control, high centrality node risks, and cost associated with the distance between nodes. The proposed solution has been evaluated with good results against a series of simulations based on smart cities topologies and configurations. Meaning in advance, we can attempt to secure these vulnerable nodes that expose the network to risk or identify if alternative links need to be established before failures occur.

Analysis of these early results, suggest an evolutionary approach is practical for optimising relatively small networks in a small number of steps. The future work will extend this methodology, applying the principals outlined to a larger physical NGI environment generating greater graphs and data sets. We are also interested in deploying the framework in a distributed format across NGI infrastructure, and will be examining associated issues with deployment. In addition, we aim to examine different approaches to optimising NGI environments, and will evaluate the effects and differences between optimising sections of an NGI instead of optimising the NGI topology as a whole. This will allow us to analyse and ascertain the most effective approach to NGI optimisation and the methodologies applications within these environments.

References

- Abdelouahid, R.A., Marzak, A., (2018), "Towards a New Interoperability Quality Model for IoTs," Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), Amman, 2018, pp. 1-6, 31 October - 1 November.
- Ai, J., Chen, H., Guo, Z., Cheng, G., Baker, T., (2019), "Mitigating Malicious Packets Attack via Vulnerability-aware Heterogeneous Network Devices Assignment," *Future Generation Computer Systems*.
- Aia, J., Chena, H., Guo, Z., Cheng, G., Baker, T., (2019), "Mitigating malicious packets attack via vulnerability-aware heterogeneous network devices assignment," *Journal of Future Generation Computer Systems*.
- Albert, R., Jeong, H., Barabasi, A.L., (2000), "Error and attack tolerance of complex networks," *Nature* 406, pp. 378.
- Ali, Z., Abbas, Z.H., Li, F.Y., (2016), "A Stochastic Routing Algorithm for Distributed IoT with Unreliable Wireless Links," *IEEE 83rd Vehicular Technology Conference (VTC Spring)*, Nanjing, pp. 1-5.
- Alfarhan, F., Alsohaily, A., (2017), "Self-organizing wireless network parameter optimization through mixed integer programming," *2017 IEEE International Conference on Communications (ICC)*, Paris, France, 2017, pp. 1-6.
- Aniketos, (2017), Aniketos deliverables, [online] <http://www.aniketos.eu/deliverables> (accessed February 2017).
- Belguith S., Kaaniche N., Mohamed M., Russello G., (2018), "Coop-DAAB: Cooperative Attribute Based Data Aggregation for Internet of Things Applications," In: Panetto H., Debruyne C., Proper H., Ardagna C., Roman D., Meersman R. (eds) *On the Move to Meaningful Internet Systems, OTM 2018 Conferences, OTM 2018, Lecture Notes in Computer Science*, vol. 11229, Springer, Cham.
- Bellingeri, M., Cassi, D., (2018), "Robustness of weighted networks," *Physica A*, 489, pp. 47 - 55.
- Blum, C., (2005), "Ant colony optimization: Introduction and recent trends," *Physics of Life Reviews*, vol. 2, no. 4, pp. 353-373, December.
- Boccaletti, S., Buldu, J., Criado, R., Flores, J., Latora, V., Pello, J., Romance, M. (2007), "Multiscale vulnerability of complex networks,"

Chaos 17, 043110.

Brownlee, J., (2011), *Clever Algorithms, Nature-Inspired Programming Recipes*, First Edition.

Brummitt, C.D., D'Souza, R.M., Leicht, E.A., (2012), "Suppressing cascades of load in interdependent networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 109, no. 12, 21 February.

Buldyrev, S.V., Parshani, R., Paul, G., Stanley, H.E., Havlin, S., (2010), "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, pp. 1025–1028.

Cadini, F., Agliardi, G.L., Zio, E., (2017), "A modeling and simulation framework for the reliability/availability assessment of a power transmission grid subject to cascading failures under extreme weather conditions," *Applied Energy*, vol. 185, part 1, pp. 267-279, 1 January.

Cai, Y., Cao, Y., Li, Y., Huang, T., Zhou, B., (2016), "Cascading Failure Analysis Considering Interaction Between Power Grids and Communications Networks," in *IEEE Transactions on Smart Grid*, vol. 7, no. 1, pp. 530-538, January.

Carmi, S., Havlin, S., Kirkpatrick, S., Shavitt, Y., Shir, E., (2007), "A model of internet topology using k-shell decomposition," in *Proceedings of the National Academy of Sciences* 104 (27). pp. 11150–11154, August.

Chen, Z., Wu, J., Xia, Y., Zhang, X., (2017) "Robustness of interdependent power grids and communication networks: a complex network perspective," *IEEE Transactions on Circuits and Systems II: Express Briefs* 99, pp. 115-119, 18 May.

Datta, A., Derek, A., Mitchell, J.C., (2004), "Secure Protocol Composition", *Electronic Notes in Theoretical Computer Science*, vol. 83, pp. 201-226.

Du, W.B., Zhou, X.L., Lordan, O., Wang, Z., Zhao, C., Zhu, Y.B., (2016) "Analysis of the Chinese Airline Network as multi-layer networks," *Transportation Research Part E: Logistics and Transportation Review*, vol. 89, pp. 108–116.

Du, W., Liang, B., Yan, G., Lordan, O., Cao, X., "Identifying vital edges in Chinese air route network via memetic algorithm," *China Journal Aeronautics* Volume 30, iss. 1, pp 330–336.

EPSRC, (2017), *Secure Component Composition for Personal Ubiquitous Computing*, Engineering and Physical Sciences Research Council [online] <http://gow.epsrc.ac.uk/NGBOViewGrant.aspx?GrantRef=GR/S01634/01> (accessed February 2017).

Farooqi, A.H., Khan, F.A., (2017), "Securing wireless sensor networks for improved performance in cloud-based environments," *Annals of Telecommunications*, pp. 1-18, 22 March.

Feng, Y., Sun, B., Zeng, A., (2017), "Cascade of links in complex networks," *Physics Letters A*, vol. 381, iss. 4, pp. 263-269, 30 January.

FIRST.Org, (2019), "CVSS, Common Vulnerability Scoring System v3.0: Specification Document" [online] <https://www.first.org/cvss/cvss-v30-specification-v1.7.pdf> (accessed May 2019).

Freeman, L.C., (1977), "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35-41.

Freeman, L.C., (1979), "Centrality in social networks conceptual clarification, *Social Networks*, vol. 1, iss. 3, pp. 215-239.

Gao, Q., Holding, D.J., Blow, K.J., (2002), "Energy Efficiency Design Challenge in Sensor Networks," in *Proceedings of London Communications Symposium 2002*.

Glissa, G., Rachedi, A., Meddeb, A., (2016), "A Secure Protocol Based on RPL for Internet of Things," *IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, pp. 1-7.

Goldshstein, V., Koganov, G. A., Surdutovich, G.I. (2004), *Vulnerability and Hierarchy of Complex Networks*, cond-mat/0409298.

Grefenstette, J.J., (1986), "Optimization of Control Parameters for Genetic Algorithms", *IEEE Transactions on Systems, man, and Cybernetics*, vol. 16, no. 1, pp. 122-128.

Hinz, O., Skiera, B., Barrot, C., Becker, J.U., (2011), "Seeding strategies for viral marketing: An empirical comparison", in *Journal of Marketing* 75, pp. 55–71.

Holme, P., Kim, B.J., Yoon, C.N., Han, S.K., (2002), "Attack vulnerability of complex networks," in *Physics Review*, vol. 65, iss. 5.

Hwang, W., Cho, Y.R., Zhang, A., Ramanathan, M., (2006), "Bridging centrality: identifying bridging nodes in scale-free networks", in *Proceedings of the 12th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 20-23, March.

Jiang, X.Y., Ma, J.F., Shen, Y.L., Zeng, Y., (2016), "Effects of link-orientation methods on robustness against cascading failures in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 457, 1–7.

Jiang, Z.Y., Liu, Z.Q., Liu, X., Ma, J.F. (2018), "Cascade phenomenon against subsequent failures in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 499, pp. 472-480.

Jina, W.X., Song, P., Liua, G.z., Stanley, E., (2015), "The cascading vulnerability of the directed and weighted network:", *Physica A: Statistical Mechanics and its Applications*, vol. 427, pp. 302-325.

Kaur, G., (2016), "A preventive approach to mitigate the effects of gray hole attack using genetic algorithm," *2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA)* (Spring), pp. 1-8, Dehradun.

- Keckemeti, G., Casale, G., Jha, D.N., Lyon, J., Ranjan, R., (2017), "Modelling and Simulation Challenges in Internet of Things," in IEEE Cloud Computing, vol. 4, no. 1. pp. 62-69, Jan-Feb.
- Khali, N., Abid, M.R., Benhaddou, D., Gerndt, M., (2014), "Wireless Sensors Networks for Internet of Things," 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Singapore, pp. 1-6.
- Kim, H., Song, J., (2013), "Social network analysis of patent infringement lawsuits," Technological Forecasting and Social Change, vol. 80, iss. 5, Pages 944-955, June.
- Lekha, D.S., Balakrishnan, K., (2018), "Attack Vulnerability of Complex Networks in Center-Based Strategies," ArXiv, 12 Dec 2018.
- Liu, R.R., Eisenberg, D.A., Seager, T.P., Lai, Y.C., (2011), "The "weak" interdependence of infrastructure systems produces mixed percolation transitions in multilayer networks," Scientific Reports 8, 2111.
- Liu, R.R., Li, M., Jia, C.X., (2016), "Cascading failures in coupled networks: The critical role of node-coupling strength across networks," Scientific Reports 6, pp. 35352.
- Liu, Y.Y., Slotine, J.J., Barabasi, A.L., (2011), "Controllability of complex networks," Nature, vol. 473, no. 7346, pp. 167-173.
- Lu, L., Chen, D., Ren, X., Zhang, Q., Zhang, Y., Zhou, T. (2016), "Vital nodes identification in complex networks," Physics Reports, vol. 650, pp. 1-63.
- Meghanathan, N., ed., (2016), Advanced methods for complex network analysis, IGI Global.
- Meland, P.H., (2011), "Service Injection: A Threat to Self-Managed Complex Systems," 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, Sydney, NSW, pp. 1-6.
- Mishkovski, I., Biey, M., Kocarev, L., (2011), "Vulnerability of complex networks," Elsevier Communications in Nonlinear Science and Numerical Simulation, vol. 16, iss. 1, January, pp. 341-349.
- Monteiro, M.S.R., Fontes, D.B.M.M., Fontes, F.A.C.C., (2013), "Concave minimum cost network flow problems solved with a colony of ants," Journal of Heuristics, vol. 19, iss. 1, pp. 1-33, February.
- Networkd, (2020), "Smart Networks in the context of NGI," Strategic Research and Innovation Agenda 2021-27, [online] <https://www.networkd2020.eu/wp-content/uploads/2018/11/networkd2020-5gia-sria-version-2.0.pdf> (accessed December 2019).
- Nie, T., Guo, Z., Zhao, K., Lu, Z. (2015), "New attack strategies for complex networks," Physica A: Statistical Mechanics and its Applications, vol. 424, April, pp. 248-253.
- Nvd.nist.gov, (2019), NVD – Home [online] <https://nvd.nist.gov/> (accessed May 2019).
- Olivas, F., Valdez, F., Castillo, O., Gonzalez, C.I., Martinez, G., (2017), "Ant colony optimization with dynamic parameter adaption based on internal type-2 fuzzy logic systems," Applied Soft Computing, vol. 53, pp. 74-87, April.
- Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F., Krmer, B.J., (2006), "Service-oriented computing: A research roadmap," in Service Oriented Computing (SOC) ser. Dagstuhl Seminar Proceedings 2006.
- Pua, C., Lia, S., Michael, A., Yanga, J., (2015), "Iterative path attacks on networks," Physics Letter A, vol. 379, iss. 28-29, pp. 1633-1638.
- Qui, T., Chen, N., Li, K., Qiao, D., Fu, Z., (2017), "Heterogeneous ad hoc networks: Architectures, advances and challenges," Ad Hoc Networks, vol. 55, pp. 143-152, February.
- Rullo, A., Midi, D., Serra, E., Bertino, E., (2017), "A Game of Things: Strategic Allocation of Security Resources for IoT," 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), Pittsburgh, PA, pp. 185-190.
- Sabidussi, G., (1966), "The centrality index of a graph," Psychometrika 31, vol. 31, iss. 4, pp. 581-603.
- Sanka, S., Hota, C., Rajarajan, M., (2010), "Secure Data Access in Cloud Computing," 2010 IEEE 4th International Conference on Internet Multimedia Services Architecture and Application, Bangalore, pp. 1-6.
- Sestini, F., Dore, J., Martinez, L., Tselentis, G., (2018), "Next Generation Internet," European Commission 2018, [online] https://ec.europa.eu/futurium/en/system/files/ged/ngi_bcn_digital_sovereignty.pdf (accessed December 2019).
- Tariq, N., Asim, M., Al-Obeidat, F., Farooqi, M., Baker, T., Hammoudeh, M., Ghafir, I., (2019), "The Security of Big Data in Fog-Enabled IoT Applications Including Blockchain: A Survey," Sensors, vol. 19.
- Walker-Roberts, S., Hammoudeh, M., Dehghantanha, A., (2018), "A Systematic Review of the Availability and Efficacy of Countermeasures to Internal Threats in Healthcare Critical Infrastructure," in IEEE Access, vol. 6, pp. 25167-25177.
- Wu, J., Zeng, J., Chen, Z., Chi, K.T., Chen, B., (2018), "Effects of traffic generation patterns on the robustness of complex networks," Physica A: Statistical Mechanics and its Applications, vol. 492, pp. 871-877.
- Wu, J., Fang, B., Fang, J., Chen, X., Chi, K.T., (2019), "Sequential topology recovery of complex power systems based on reinforcement learning," Physica A: Statistical Mechanics and its Applications, vol. 535, 122487.
- Wu, P., Wang, M., (2018), "Routing Algorithm based on Energy and Hop Number for Linear Distributed WSN," in IEEE 7th Data

Driven Control and Learning Systems Conference (DDCLS), Enshi, pp. 194-198, November.

Yan, X., Zhang, L., Wu, Y., Luo, Y., Zhang, X., (2017), "Secure smart grid communications and information integration based on digital watermarking in wireless sensor networks," *Enterprise Information Systems*, vol. 11, iss. 2.

Yao, H., Yang, H., Zhang, A., Fang, C., Guo, Y., (2017), "WLAN interference self-optimization using SOM Neural Networks," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 3, 10 February.

Yi, L., Tong, X., Wang, Z., Zhang, M., Zhu, H., Liu, J., (2019), "A Novel Block Encryption Algorithm Based on Chaotic S-Box for Wireless Sensor Network," in *IEEE Access*, vol. 7, pp. 53079-53090, April.

Zhang, K., Ni, J., Yang, K., Liang, X., Ren, J., Shen, X.S., (2017), "Security and Privacy in Smart City Applications: Challenges and Solutions," in *IEEE Communication Magazine*, vol. 55, no. 1, pp. 122-129, January.

Zhong-Yuan, JiangYong, ZengZhi-Hong, LiuJian-FengMa, (2019), "Identifying critical nodes' group in complex networks," *Physica A: Statistical Mechanics and its Applications*, vol. 514, pp. 121-132.

Zhou, B., Arabo, A., Drew, O., Llewellyn - Jones, D., Merabti, M., Shi, Q., Waller, A., Craddock, R., Jones, G., Arnold, K.L.Y., (2008), "Data Flow Security Analysis for System-of-Systems in a Public Security Incident," in *The 3rd Conference on Advances in Computer Security and Forensics (ACSF 2008)*, Liverpool, UK, 10 - 11 July.