Erol Gelenbe · Marija Jankovic ·
Dionysios Kehagias · Anna Marton ·
Andras Vilmos (Eds.)

Communications in Computer and Information Science 1596

Security in Computer and Information Sciences

Second International Symposium, EuroCybersec 2021
Nice, France, October 25–26, 2021
Revised Selected Papers

Springer

Communications in Computer and Information Science 1596

More information about this series at https://link.springer.com/bookseries/7899

Erol Gelenbe · Marija Jankovic ·
Dionysios Kehagias · Anna Marton ·
Andras Vilmos (Eds.)

Security in Computer and Information Sciences

Second International Symposium, EuroCybersec 2021
Nice, France, October 25–26, 2021
Revised Selected Papers

Springer

Editors

Erol Gelenbe
IITIS-PAN
Gliwice, Poland

Marija Jankovic
ITI-CERTH
Thessaloniki, Greece

Dionysios Kehagias
ITI-CERTH
Thessaloniki, Greece

Anna Marton
Safepay Systems
Budapest, Hungary

Andras Vilmos
Safepay Systems
Budapest, Hungary

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-3-031-09356-2 ISBN 978-3-031-09357-9 (eBook)
https://doi.org/10.1007/978-3-031-09357-9

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The Second International ISCIS Symposium on Security in Computer and Information Sciences (EuroCybersec 2021) was held in Nice, France, during October 25–26, 2021. It was supported by the European Commission H2020 Program through the IoTAC Research and Innovation Action, under Grant Agreement No. 952684.

The symposium was organized by the European Union's IoTAC project and the Institute of Theoretical and Applied Informatiocs of the Polish Academy of Sciences (IITIS-PAN), based on an open call for papers and presentations selected by the Program Committee.

After the oral presentations, the Program Committee then reviewed the papers that were presented to evaluate once again their originality, scientific quality, and technical maturity, and made a further selection resulting in the nine papers included in these proceedings. All papers coauthored by committee members were handled in an appropriate review process.

The key areas covered by these proceedings include the Internet of Things (IoT), cybersecurity, IoT gateways, IoT attack detection and mitigation, the IoT massive access problem, adaptive routing for security, quality of service (QoS), and energy optimization in Fog and Edge systems that support the IoT.

EuroCybersec 2021 follows up on a previous workshop held in 2018 at Imperial College London, UK [15], as part of the sequence of International Symposia on Computer and Information Sciences (ISCIS) that started in 1986 and have been held over the years in Turkey, France, the USA, the UK, and Poland [2, 7–9, 14, 18–20, 37].

The ease of access to the Internet with a very high traffic yet inexpensive business model has raised major concerns with regard to cybersecurity, since the Internet offers low cost high volume access not only to legitimate users but also to various malicious users. The advent of IoT has thus created even more opportunities to attack not just virtual facilities but also cyber-physical systems [24].

Of course, various organizations, including the European Union, have published recommendations for Internet security and privacy [12], but this has by no means reduced the number of cyberattacks over the years. This growing insecurity in systems and networks also results in increased energy consumption by ICT [16, 34] due to increased traffic as well as more software that is meant to insure secure operation. These concerns also raise major issues that combine performance and QoS, security, and energy consumption [27].

As a consequence, the European Commission has increasingly supported research projects in these fields [1], with projects such as NEMESYS on the cybersecurity of mobile telephone systems [3, 33], SDK4ED on energy savings in dependable and secure systems [35, 36], KONFIDO [10, 11, 31, 32] on the security of health informatics systems, GHOST [5, 6, 26] regarding the security of IoT home gateways, and SerIoT on the cybersecurity of IoT systems[4, 13].

The current project IoTAC [25] pursues this work and aims at securing IoT networks by protecting IoT gateways using techniques such as Botnet detection and system wide vulnerability assessement [28, 29], disruptive checkpoints, and assuring efficient massive IoT device access to gateways [21, 23]. The topics covered by the EuroCyberSec 2021 symposium reflected the aims of the IoTAC project.

Over 20 paper presentations were submitted of which 15 were retained for the symposium, and nine full papers were selected for these proceedings by the Program Committee based on technical quality. One additional review paper was invited. Over 40 participants attended, with some 15 physically present and the remainder online.

The papers in these proceedings discuss aspects specifically relevant to the IoTAC project, and also of broad interest to cybersecurity and related European Union projects, including other research projects and some of their outcomes, such as the combined societal and technical implications of IoT cybersecurity.

Since software is the ultimate target of cyberattacks, software vulnerability detection methods were examined by examining the influence of the "vocabulary" used inside programs, relating to the security by design for software systems considered in IoTAC. Signal processing techniques applied to digital data on the internal computer data transfer "bus" can help detect anomalies or attacks on servers, while incremental attack detection can also be performed, with ongoing learning and detection occurring as the packet traffic flows into a gateway [30].

The important issue of energy consumption for battery powered drone surveillance missions, in order to optimize actions within a mission and maximize mission duration, is of great importance in both civilian and military applications, and is also relevant to one of the IoTAC use cases involving Airbus industries.

The authentication of IoT devices by hardware and software means with a hybrid approach connects us to the SETIT project, also funded by the European Commission. Secure authentication schemes are also discussed, as well as fast adaptive routing-based methods aimed at reducing energy consumption while improving both performance and system security at the Edge [22].

IoT also has a massive access problem when a large number of IoT devices access a gateway frequently or periodically, which can be addresed by novel traffic shaping techniques [17].

We hope that you find these papers interesting and fruitful for your own research.

Erol Gelenbe
Marija Jankovic
Dionysios Kehagias
Anna Marton
Andras Vilmos

References

1. https://www.grantsoffice.com/Portals/0/funded/issues/FUNDEDOct2021.pdf
2. Abdelrahman, O.H., Gelenbe, E., Görbil, G., Lent, R. (eds.): Information Sciences and Systems 2015 - 30th International Symposium on Computer and Information Sciences, ISCIS 2015, London, UK, 21–24 September 2015, Lecture Notes in Electrical Engineering, vol. 363. Springer (2016)
3. Abdelrahman, O.H., Gelenbe, E., Görbil, G., Oklander, B.: Mobile network anomaly detection and mitigation: The NEMESYS approach. In: Information Sciences and Systems 2013, pp. 429–438. Springer (2013)
4. Baldini, G., et al.: Iot network risk assessment and mitigation: the seriot approach. In: Soldatos, J. (ed.) Security Risk Management for the Internet of Things: Technologies and Techniques for IoT Security, Privacy and Data Protection. pp. 87–104. NOW Publishers (2020). https://doi.org/10.1561/9781680836837, https://www.nowpublishers.com/article/Chapter/9781680836820?cId=978-1-68083-683-7.ch5
5. Brun, O., Yin, Y., Gelenbe, E.: Deep learning with dense random neural network for detecting attacks against IoT-connected home environments. Procedia Computer Science **134**, pp. 458–463 (2018)
6. Collen, A., et al.: GHOST - safe-guarding home IoT environments with personalised real-time risk control. In: Gelenbe, E., Campegiani, P., Czachorski, T., Katsikas, S., Komnios, I., Romano, L., Tzovaras, D. (eds.) Recent Cybersecurity Research in Europe: Proceedings of the 2018 ISCIS Security Workshop, Imperial College London. Communications in Computer and Information Science, vol. 821, pp. 68–78. Springer (2018)
7. Czachórski, T., Gelenbe, E., Grochla, K., Lent, R. (eds.): Computer and Information Sciences - 31st International Symposium, ISCIS 2016, Kraków, Poland, October 27–28, 2016, Proceedings, Communications in Computer and Information Science, vol. 659. Springer (2016)
8. Czachórski, T., Gelenbe, E., Grochla, K., Lent, R.: Computer and Information Sciences - 32nd International Symposium, ISCIS 2018, held at the 24th IFIP World Computer Congress, WCC 2018, Poznan, Poland, September 20–21, 2018, Proceedings (2018)
9. Czachórski, T., Gelenbe, E., Lent, R. (eds.): Information Sciences and Systems 2014 - Proceedings of the 29th International Symposium on Computer and Information Sciences, ISCIS 2014, Krakow, Poland, October 27–28, 2014. Springer (2014)
10. Diamantopoulos, S., et al.: Secure cross-border exchange of health related data: The KONFIDO approach. In: International Conference on Internet and Distributed Computing Systems, pp. 318–327. Springer (2019)
11. Diamantopoulos, S., et al.: Secure cross-border exchange of health related data: The KONFIDO approach. In: EDCC, pp. 73–74. IEEE (2019)
12. European Commission Cybersecurity Policies: https://digital-strategy.ec.europa.eu/en/policies/cybersecurity-policies
13. Frötscher, A., Monschiebl, B., Drosou, A., Gelenbe, E., Reed, M.J., Al-Naday, M.: Improve cybersecurity of C-ITS road side infrastructure installations: the SerIoT - secure and safe IoT approach. In: 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), pp. 1–5. IEEE (2019)
14. Gelenbe, E.: The 24th International Symposium on Computer and Information Sciences, ISCIS 2009, 14–16 September 2009, North Cyprus. IEEE (2009)
15. Gelenbe, E., et al.: Security in computer and information sciences: First International ISCIS Security Workshop, Euro-Cybersec 2018, London, UK, February 26–27, 2018, revised selected papers (2018)

16. Gelenbe, E., Caseau, Y.: The impact of information technology on energy consumption and carbon emissions. Ubiquity 2015 (June), pp. 1–15. ACM (2015)
17. Gelenbe, E., Czachorski, T., Marek, D., Nakıp, M.: Mitigating the massive access problem in the internet of things. In: EuroCybersec 2021. Springer (2022)
18. Gelenbe, E., Lent, R. (eds.): Computer and Information Sciences III - 27th International Symposium on Computer and Information Sciences, Paris, France, October 3–4, 2012. Springer (2013)
19. Gelenbe, E., Lent, R. (eds.): Information Sciences and Systems 2013 - Proceedings of the 28th International Symposium on Computer and Information Sciences, ISCIS 2013, Paris, France, October 28–29, 2013, Lecture Notes in Electrical Engineering, vol. 264. Springer (2013)
20. Gelenbe, E., Lent, R., Sakellari, G., Sacan, A., Toroslu, I.H., Yazici, A.: Computer and Information Sciences - Proceedings of the 25th International Symposium on Computer and Information Sciences, London, UK, September 22–24, 2010, Lecture Notes in Electrical Engineering, vol. 62. Springer (2010).
21. Gelenbe, E., Nakip, M., Marek, D., Czachorski, T.: Diffusion analysis improves scalability of IoT networks to mitigate the massive access problem. In: IEEE MASCOTS 2021: 29th International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 1–6. IEEE (2021)
22. Gelenbe, E., Nowak, M.P., Frohlich, P., Fiołka, J., Chęciński, J.: Energy, QoS and security aware services at the edge. In: EuroCybersec 2021. Springer (2022)
23. Gelenbe, E., Sigman, K.: IoT traffic shaping and the massive access problem. In: ICC 2022, IEEE International Conference on Communications, 16–20 May 2022, Seoul, South Korea. pp. 2290–2295 (2022)
24. Gelenbe, E., Wu, F.J.: Future research on cyber-physical emergency management systems. Future Internet **5**(3), pp. 336–354 (2013)
25. Siavvas M., et al.: The IoTAC software security-by-design platform: Concept, challenges, and preliminary overview. In: DRCNN 2022: 1st International Workshop on Key challenges in Global Cybersecurity: Efforts and Trends in EU (KCYEU), pp. 1–6. IEEE (2022)
26. Kadioglu, Y.M., Gelenbe, E.: Product-form solution for cascade networks with intermittent energy. IEEE Syst. J. **13**(1), pp. 918–927. IEEE (2019)
27. Kehagias, D.D., Jankovic, M., Siavvas, M.G., Gelenbe, E.: Investigating the interaction between energy consumption, quality of service, reliability, security, and maintainability of computer systems and networks. SN Comput. Sci. **2**(1), 23 (2021)
28. Nakip, M., Gelenbe, E.: Mirai botnet attack detection with auto-associative dense random neural networks. In: 2021 IEEE Global Communications Conference. vol. 2021, pp. 1–6. IEEE (2021)
29. Nakip, M., Gelenbe, E.: Randomization of data generation times improves performance of predictive IoT networks. In: IEEE World Forum on Internet of Things (WF IoT), July 14–21, 2021. p. 5161. IEEE (2021)
30. Nakip, M., Gelenbe, E.: Botnet attack detection with incremental online learning. In: EuroCybersec 2021. Springer (2022)

31. Nalin, M., et al.: The European cross-border health data exchange roadmap: Case study in the Italian setting. Journal of Biomedical Informatics **94**, 103183 (2019)
32. Natsiavas, P., et al.: Developing an infrastructure for secure patient summary exchange in the eu context: Lessons learned from the KONFIDO project. Health Informatics Journal **27**(2), 14604582211021459 (2021)
33. Pavloski, M., Gelenbe, E.: Mitigating for signalling attacks in UMTS networks. In: Information Sciences and Systems 2014, pp. 159–165. Springer (2014)
34. Pernici, B., Aiello, M., Vom Brocke, J., Donnellan, B., Gelenbe, E., Kretsis, M.: What IS can do for environmental sustainability: a report from CAiSE'11 panel on green and sustainable IS. Communications of the Association for Information Systems **30**(1), 18 (2012)
35. Siavvas, M., et al.: An empirical evaluation of the relationship between technical debt and software security. In: ICIST 2019 Proceedings. vol. 1, pp. 199–203 (2019)
36. Siavvas, M.G., Gelenbe, E.: Optimum interval for application-level checkpoints. In: CSCloud/EdgeCom. pp. 145–150. IEEE (2019)
37. Tugcu, T., Caglayan, M.U., Alagoz, F., Gelenbe, E. (eds.): New Trends in Computer Networks: 20th International Symposium on Computer and Information Sciences (2005)

Organization

Organizing Committee

Erol Gelenbe	IITIS, Polish Academy of Sciences, Poland
Marija Jankovic	ITI-CERTH, Greece
Dionysios Kehagias	ITI-CERTH, Greece
Anna Marton	SafePay, Hungary
Andras Vilmos	ATOS, Hungary

Program Committee

Erol Gelenbe (Chair)	IITIS, Polish Academy of Sciences
Levente Buttyan	Budapest University of Technology and Economics, Hungary
Ufuk Caglayan	Yasar University, Turkey
Maria Carla Calzarossa	University of Pavia, Italy
Tadeusz Czachorski	IITIS, Polish Academy of Sciences, Poland
Cuneyt Guzelis	Yasar University, Turkey
Peter Hoffman	T-Sec, Germany
Marija Jankovic	ITI-CERTH, Greece
Dionysios Kehagias	ITI-CERTH, Thessaloniki, Greece
Ioannis Mavridis	University of Macedonia, Greece
Miltiadis Siavvas	ITI-CERTH, Greece

Contents

AI and Quality of Service Driven Attack Detection, Mitigation and Energy Optimization: A Review of Some EU Project Results

Mehmet Ufuk Çağlayan(✉)

Department of Computer Engineering, Yaşar University, Bornova, Izmir, Turkey
ufuk.caglayan@yasar.edu.tr

Abstract. This article summarizes briefly the contributions presented in this EuroCyberSecurity Workshop 2021 which is organized as part of the series of International Symposia on Computer and Information Sciences (ISCIS), with the support of the European Commission funded IoTAC Project, that was held on November and in NIce, France, and sponsored by the Institute of Teoretical and Applied Informatics of the Polish Academy of Sciences. It also summarizes some of the research contributions of several EU Projects including NEMESYS, GHOST, KONFIDO, SDK4ED and IoTAC, primarily with a cybersecurity and Machine Learning orientation. Thus subjects covered include the cybersecurity of Mobile Networks and of the Internet of Things (IoT), the design of IoT Gateways and their performance, the security of networked health systems that provide health services to individuals across the EU Member states, as well as the issues of energy consumption by ICT which are becoming increasingly important, including in the cybersecurity perspective, as we focus increasingly on climate change and the needed transition towards highly reduced emissions. Many of the techniques and results discussed in this article are based either on Machine Learning (ML) methods, or on methods for the performance modeling and optimization of networked and distributed computer systems.

Keywords: Internet of Things (IoT) · Cybersecurity · Secure mobile networks · IoT gateways · Secure health informatics · Attack detection · IoT massive access problem · Attack mitigation · Adaptive routing · ICT energy optimization

1 Introduction

The International Symposia on Computer and Information Sciences (ISCIS) were started in 1986 in Turkey by Erol Gelenbe, and held in Turkey, France, the USA, the UK, and Poland with proceedings [4,14–16,38,57,58,60,112] including a wide range of topics published by Springer.

E. Gelenbe et al. (Eds.): EuroCybersec 2021, CCIS 1596, pp. 1–12, 2022.
https://doi.org/10.1007/978-3-031-09357-9_1

This ISCIS CyberSecurity 2021 Symposium that was held in Nice, France, as part of this series, specializes for the second time on Cybersecurity following a previous event [45], which is my main research interest [6,21,90]. Indeed, Cybersecurity is at the forefront of serious technical issues in Computer Science as we transition to highly inter-dependent cyber-physical systems [76], and the European Union published its recommendation for security and privacy [22]. Furthermore, insecurity in systems and networks and the techniques that are used to defend our systems, are also increasing energy consumption in computer systems and network and their CO_2 impact, and the costs of operating them [27,46,86]. Hence energy consumption in mobile network has also received attention [3,42].

Thus the European Commission funded research projects in this field have significantly increased [1] over recent years and this introduction summarizes related research undertaken throughout Europe and includes five recent EC funded projects:

- NEMESYS on the cybersecurity of mobile telephone system [5,52,53,82,101],
- The project SDK4ED that mainly focused on energy savings [87,103] but also considered issues of Cybersecurity and Reliability [109].
- KONFIDO [17,18,96,97] on the security of communications and data transfers for interconnected European national or regional health services,
- GHOST [8,11] regarding the security of IoT systems for the home, and the design of secure IoT home gateways,
- SerIoT on the Cybersecurity of IoT systems [7,31] with a range of applications in supply chains, smart cities, smart manufacturing, and other areas.
- IoTAC, which aims at securing IoT networks by strengthening the protection of gateways using novel techniques such as Botnet detection, system wide vulnerability assessment [93,94], disruptive checkpoints, and assuring the optimization of the massive access to IoT gateways [67,75].

It also discusses some results from the SDK4ED project concerning the energy efficient handling of system reliability issues through checkpointing [107,108].

2 Improving the Security of Mobile Telephony

Cybersecurity of mobile telephony is a fundamental societal issue. The related problems are exacerbate by the fact that most mobile phones offer opportunistic connections [84,85] to WIFI and other wireless networks which are not part of the mobile operators' core infrastructure. This creates vulnerabilities that need to be monitored on the mobile device itself, which is the motivation for the work in [26,81].

On the other hand, the work described in [2,102], concerns a form of Distributed Denial of Service (DDoS) attacks on the signalling plane of the core mobile network which are caused by malicious software which is deposited in the mobile devices. Related work conducted within the EU NEMESYS project [41,43,83] using queueing theoretic methods [25,34].

Early work on DDoS Attacks [65] had proposed self-aware networks and the Cognitive Packet Network (CPN) [39,77,80] to detect and counter-attack against DDoS, by identifying sources of attacks by following upstream the attacking traffic, using CPN's ACK packets to "drop" attacking traffic at upstream routers [65,100]. It was also applied to mitigate worm attacks and to deviate user traffic so as to avoid insecure nodes [37,104,105]. Related issues include the management of keys [114,115], and the study and mitigation of signalling storms in mobile telephony [26,102].

3 Security of the Trans-European Health Informatics Network

Large numbers of travellers from one European country to another sometimes need to access health services in the country they are visiting. These health services are typically based on a national model, or a regional model inside a given country such as Italy. Thus the KONFIDO project addressed the important issue of providing a secure support to European health systems.

The corresponding informatics systems, with their patient data bases are also nationally or regionally based, so that when the medical practitioner in one country or region is required to diagnose and treat a visitor from some other region or country, she/he will need to access the patient's data remotely. KONFIDO's aim is to improve the cybersecurity of such systems, while improving also their inter-operability across countries and regions in Europe.

Thus the work in [111] presents an overall view and challenges of the project, while in [98] the authors present an analysis of the corresponding user requirements. Such systems have obvious performance optimization issues which are discussed in [72]. Keeping track of the transactions in such a system through blockchains is suggested in [9].

4 Contributions to the Security of the IoT

To exploit the value that the IoT generated provides requires the protection of privacy and in many cases data will have to be rendered strongly anonymous. It will also require specific security not just for the IoT devices and networks, but also for the IoT data repositories in the Cloud and their access networks. These aspects are complicated by the simplicity of many IoT devices which cannot be integrated in complex distributed communication infrastructures that would require communications to be synchronized or schedules [10,74].

Thus in [11] an overview of the principles and achievements of the GHOST project are presented, which started in May of 2017 and which ran for three years. The project addressed safe-guarding home IoT environments through appropriate software that can be installed on home IoT gateways, and it also creates a prototype and test-bed using specific equipment from the TELEVES company.

Related to this project, machine learning methods were developed for the detection of network attacks on IoT gateways [8] based on Deep Learning

[78,79,106] with the Random Neural Network [32,33,35,54] and its extensions [89]. Related to the GHOST project, other recent work discusses the effect and mitigation of attacks on the batteries which supply the power of many lightweight IoT network nodes [55].

The SerIoT project that was started in 2018 [19] also produced valuable results [48]. Its technical scope included SerCPN [29,30], a specific secure network [49] for managing geographically distributed IoT devices and services using the principles of the Cognitive Packet Network (CPN) tested in several experiments [28,59,61,62,64]. CPN uses "Smart" Packets (SPs) to search for paths and measure QoS while the network is in operation, via Reinforcement Learning using a Random Neural Network, and based on the QoS Goal pursued by the end user. When an SP reaches its destination, its measurements are returned by an ACK packet to the intermediate nodes of the path that was identified by the SP, and to the end user, providing the QoS offered by the path that the SP travelled. Source nodes receive ACKs and take the decision to switch to the path that offers the best security or quality of service [50,51,56,63].

Extensions with a genetic algorithm [36] was also tested [92]. An interesting development in SerIoT combines energy aware routing [40,66] and security, and admission control [73]. Adaptive techniques for wireless IoT traffic to achieve better QoS are also found in [68–70,99] and summarized in [20,91], while the RNN with adaptive approaches was shown to offer opportunities for massive video compression [12,13], as well as for managing Cloud servers [113]. Such adaptive techniques that support the interaction between security metrics, performance and energy consumption were also discussed in a paper in this volume [71].

The subsequent IoTAC project has lead to incremental techniques for learning from user traffic and then testing for an attack as described in [95]. In IoTAC, there was also substantial work on dealing with severe performance issues due to the large flows of IoT packets towards gateways from thousands of IoT devices, so that the resulting Massive Access Problem (MAP) has to be mitigated with novel traffic shaping techniques [47].

5 Conclusions

The existence of frequent and effective cyberattacks on public networks and information technology infrastructures motivates education and research on Cybersecurity. The field that started with the need to encrypt data and create secure systems through strong means for security such as passwords, authentication schemes, firewalls and cryptographic keys, has now substantially evolved towards the detection and mitigation of cyberattacks. In addition issues with respect to software's own specific vulnerabilities [23] and the need to detect and mitigate such properties has also become important [23,24,44,88,109,110].

Indeed, we now realize that hoping to use static means of defence in Cybersecurity is largely ineffective unless it is accompanied by real-time techniques that rapidly react to possible malicious actions or attempts to attack a system.

Thus the field of Cybersecurity research has now entered a far broader phase with much more substantial activity. Its support through several European Union research programs demonstrates a new level of maturity that attempts to attain higher levels of performance and effectiveness through self-adaptation and system reconfiguration.

References

1. https://www.grantsoffice.com/Portals/0/funded/issues/FUNDEDOct2021.pdf
2. Abdelrahman, O.H., Gelenbe, E.: A data plane approach for detecting control plane anomalies in mobile networks. In: Mandler, B., et al. (eds.) IoT360 2015. LNICST, vol. 169, pp. 210–221. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47063-4_19
3. Abdelrahman, O.H., Gelenbe, E.: A diffusion model for energy harvesting sensor nodes. In: 2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 154–158. IEEE (2016)
4. Abdelrahman, O.H., Gelenbe, E., Gorbil, G., Lent, R. (eds.): Information Sciences and Systems 2015. LNEE, vol. 363. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-22635-4
5. Abdelrahman, O.H., Gelenbe, E., Görbil, G., Oklander, B.: Mobile network anomaly detection and mitigation: the NEMESYS approach. In: Gelenbe, E., Lent, R. (eds.) Information Sciences and Systems 2013, pp. 429–438. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-01604-7_42
6. Akgün, M., Çaglayan, M.U.: Towards scalable identification in RFID systems. Wireless Pers. Commun. **86**(2), 403–421 (2016). https://doi.org/10.1007/s11277-015-2936-7
7. Baldini, G., et al.: IoT network risk assessment and mitigation: the SerIoT approach (2020)
8. Brun, O., Yin, Y., Gelenbe, E.: Deep learning with dense random neural network for detecting attacks against IoT-connected home environments. Procedia Comput. Sci. **134**, 458–463 (2018)
9. Castaldo, L., Cinque, V.: Blockchain-based logging for the cross-border exchange of eHealth data in Europe. In: Gelenbe, E., et al. (eds.) Euro-CYBERSEC 2018. CCIS, vol. 821, pp. 46–56. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95189-8_5
10. Chesnais, A., Gelenbe, E., Mitrani, I.: On the modeling of parallel access to shared data. Commun. ACM **26**(3), 196–202 (1983)
11. Collen, A., et al.: GHOST - safe-guarding home IoT environments with personalised real-time risk control. In: Gelenbe, E., et al. (eds.) Euro-CYBERSEC 2018. CCIS, vol. 821, pp. 68–78. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95189-8_7
12. Cramer, C., Gelenbe, E., Bakircloglu, H.: Low bit-rate video compression with neural networks and temporal subsampling. Proc. IEEE **84**(10), 1529–1543 (1996)
13. Cramer, C.E., Gelenbe, E.: Video quality and traffic QoS in learning-based subsampled and receiver-interpolated video sequences. IEEE J. Sel. Areas Commun. **18**(2), 150–167 (2000)
14. Czachórski, T., Gelenbe, E., Grochla, K., Lent, R. (eds.): ISCIS 2016. CCIS, vol. 659. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47217-1
15. Czachórski, T., Gelenbe, E., Grochla, K., Lent, R. (eds.): ISCIS 2018. CCIS, vol. 935. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00840-6

16. Czachórski, T., Gelenbe, E., Lent, R. (eds.): Information Sciences and Systems 2014. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09465-6
17. Diamantopoulos, S., et al.: Secure cross-border exchange of health related data: the KONFIDO approach. In: Montella, R., Ciaramella, A., Fortino, G., Guerrieri, A., Liotta, A. (eds.) IDCS 2019. LNCS, vol. 11874, pp. 318–327. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34914-1_30
18. Diamantopoulos, S., et al.: Secure cross-border exchange of health related data: the KONFIDO approach. In: EDCC, pp. 73–74. IEEE (2019)
19. Domanska, J., Gelenbe, E., Czachorski, T., Drosou, A., Tzovaras, D.: Research and innovation action for the security of the internet of things: the SerIoT project. In: Gelenbe, E., et al. (eds.) Euro-CYBERSEC 2018. CCIS, vol. 821, pp. 101–118. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95189-8_10
20. Du, J., Jiang, C., Gelenbe, E., Zhang, H., Ren, Y.: Traffic offloading in software defined ultra-dense networks. In: Ultra-Dense Networks: Principles and Applications, p. 164 (2020)
21. Ermis, O., Bahtiyar, S., Anarim, E., Çaglayan, M.U.: A key agreement protocol with partial backward confidentiality. Comput. Netw. **129**, 159–177 (2017). https://doi.org/10.1016/j.comnet.2017.09.008
22. European Commission: Cybersecurity Policies. https://digital-strategy.ec.europa.eu/en/policies/cybersecurity-policies
23. Filus, K., Boryszko, P., Domańska, J., Siavvas, M., Gelenbe, E.: Efficient feature selection for static analysis vulnerability prediction. Sensors **21**, 1113 (2021). https://doi.org/10.3390/s21041133
24. Filus, K., Siavvas, M., Domańska, J., Gelenbe, E.: The random neural network as a bonding model for software vulnerability prediction. In: Calzarossa, M.C., Gelenbe, E., Grochla, K., Lent, R., Czachórski, T. (eds.) MASCOTS 2020. LNCS, vol. 12527, pp. 102–116. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68110-4_7
25. Fourneau, J.M., Gelenbe, E., Suros, R.: G-networks with multiple classes of negative and positive customers. Theoret. Comput. Sci. **155**(1), 141–156 (1996)
26. Francois, F., Abdelrahman, O.H., Gelenbe, E.: Feasibility of signaling storms in 3G/UMTS operational networks. In: Mandler, B., et al. (eds.) IoT360 2015. LNICST, vol. 169, pp. 187–198. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47063-4_17
27. Francois, F., Abdelrahman, O.H., Gelenbe, E.: Towards assessment of energy consumption and latency of LTE UES during signaling storms. In: Abdelrahman, O.H., Gelenbe, E., Gorbil, G., Lent, R. (eds.) Information Sciences and Systems 2015. LNEE, vol. 363, pp. 45–55. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-22635-4_4
28. Francois, F., Gelenbe, E.: Optimizing secure SDN-enabled inter-data centre overlay networks through cognitive routing. In: 2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 283–288. IEEE (2016)
29. Fröhlich, P., Gelenbe, E., Fiołka, J., Checinski, J., Nowak, M., Filus, Z.: Smart SDN management of fog services to optimize QoS and energy. Sensors **21**, 3105 (2021). https://doi.org/10.3390/s21093105
30. Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., Zurada, J.M. (eds.): ICAISC 2020. LNCS (LNAI), vol. 12415. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61401-0

31. Frötscher, A., Monschiebl, B., Drosou, A., Gelenbe, E., Reed, M.J., Al-Naday, M.: Improve cybersecurity of c-its road side infrastructure installations: the SerIoT-secure and safe IoT approach. In: 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), pp. 1–5. IEEE (2019)
32. Gelenbe, E.: Random neural networks with negative and positive signals and product form solution. Neural Comput. **1**(4), 502–510 (1989)
33. Gelenbe, E.: Stability of the random neural network model. Neural Comput. **2**(2), 239–247 (1990)
34. Gelenbe, E.: G-networks with signals and batch removal. Probab. Eng. Inf. Sci. **7**(3), 335–342 (1993)
35. Gelenbe, E.: Learning in the recurrent random neural network. Neural Comput. **5**(1), 154–164 (1993)
36. Gelenbe, E.: Genetic algorithms with analytical solution. In: Proceedings of the 1st Annual Conference on Genetic Programming, pp. 437–443. MIT Press (1996)
37. Gelenbe, E.: Dealing with software viruses: a biological paradigm. Inf. Secur. Tech. Rep. **12**(4), 242–250 (2007)
38. Gelenbe, E.: The 24th International Symposium on Computer and Information Sciences, ISCIS 2009, 14–16 September 2009. IEEE (2009)
39. Gelenbe, E.: Steps toward self-aware networks. Commun. ACM **52**(7), 66–75 (2009)
40. Gelenbe, E.: Energy packet networks: ICT based energy allocation and storage. In: Rodrigues, J.J.P.C., Zhou, L., Chen, M., Kailas, A. (eds.) GreeNets 2011. LNICST, vol. 51, pp. 186–195. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33368-2_16
41. Gelenbe, E., Abdelrahman, O.H.: Countering mobile signaling storms with counters. In: Mandler, B., et al. (eds.) IoT360 2015. LNICST, vol. 169, pp. 199–209. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47063-4_18
42. Gelenbe, E., Abdelrahman, O.H.: An energy packet network model for mobile networks with energy harvesting. Nonlinear Theory Its Appl. IEICE **9**(3), 1–15 (2018) https://doi.org/10.1587/nolta.9.1
43. Gelenbe, E., Abdelrahman, O.H., Gorbil, G.: Detection and mitigation of signaling storms in mobile networks. In: 2016 International Conference on Computing, Networking and Communications (ICNC), pp. 1–5. IEEE (2016)
44. Gelenbe, E., Boryszko, P., Siavvas, M., Domanska, J.: Optimum checkpoints for time and energy. In: 2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 1–8. IEEE (2020)
45. Gelenbe, E., et al. (eds.): Euro-CYBERSEC 2018. CCIS, vol. 821. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95189-8
46. Gelenbe, E., Caseau, Y.: The impact of information technology on energy consumption and carbon emissions. Ubiquity **2015**, 1–15 (2015)
47. Gelenbe, E., Czachorski, T., Marek, D., Nakıp, M.: Mitigating the massive access problem in the internet of things. In: Gelenbe, E., et al. (Eds.) EuroCybersec 2021, CCIS 1596, pp. 118–132. Springer, Cham (2022)
48. Gelenbe, E., Domanska, J., Czachorski, T., Drosou, A., Tzovaras, D.: Security for internet of things: the SerIoT project. In: Proceedings of the International Symposium on Networks, Computers and Communications. IEEE, June 2018
49. Gelenbe, E., Domanska, J., Frohlich, P., Nowak, M., Nowak, S.: Self-aware networks that optimize security, QoS and energy. Proc. IEEE **108**(7), 1150–1167 (2020)

50. Gelenbe, E., Gellman, M.: Can routing oscillations be good? The benefits of route-switching in self-aware networks. In: 2007 15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, pp. 343–352. IEEE (2007)
51. Gelenbe, E., Gellman, M.: Oscillations in a bio-inspired routing algorithm. In: 2007 IEEE International Conference on Mobile Adhoc and Sensor Systems, pp. 1–7. IEEE (2007)
52. Gelenbe, E., et al.: NEMESYS: enhanced network security for seamless service provisioning in the smart mobile ecosystem. In: Gelenbe, E., Lent, R. (eds.) Information Sciences and Systems 2013, pp. 369–378. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-01604-7_36
53. Gelenbe, E., et al.: Security for smart mobile networks: the NEMESYS approach. In: 2013 International Conference on Privacy and Security in Mobile Systems (PRISMS), pp. 1–8. IEEE (2013)
54. Gelenbe, E., Hussain, K.F.: Learning in the multiple class random neural network. IEEE Trans. Neural Networks **13**(6), 1257–1267 (2002)
55. Gelenbe, E., Kadioglu, Y.M.: Energy life-time of wireless nodes with network attacks and mitigation. In: Proceedings of ICC 2018, 20–24 May 2018, W04: IEEE Workshop on Energy Harvesting Wireless Communications. IEEE (2018)
56. Gelenbe, E., Lent, R.: Power-aware ad hoc cognitive packet networks. Ad Hoc Netw. **2**(3), 205–216 (2004)
57. Gelenbe, E., Lent, R. (eds.): Computer and Information Sciences III - 27th International Symposium on Computer and Information Sciences, Paris, France, 3–4 October 2012. Springer, London (2013). https://doi.org/10.1007/978-1-4471-4594-3
58. Gelenbe, E., Lent, R. (eds.): Information Sciences and Systems 2013 - Proceedings of the 28th International Symposium on Computer and Information Sciences, ISCIS 2013, Paris, France, 28–29 October 2013, Lecture Notes in Electrical Engineering, vol. 264. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-01604-7
59. Gelenbe, E., Lent, R., Montuori, A., Xu, Z.: Cognitive packet networks: QoS and performance. In: 10th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, MASCOTS 2002, Proceedings, pp. 3–9. IEEE (2002)
60. Gelenbe, E., Lent, R., Sakellari, G., Sacan, A., Toroslu, I.H., Yazici, A.: Computer and Information Sciences - Proceedings of the 25th International Symposium on Computer and Information Sciences, London, UK, 22–24 September 2010. LNEE, vol. 62. Springer, Dordrecht (2010). https://doi.org/10.1007/978-90-481-9794-1
61. Gelenbe, E., Lent, R., Xu, Z.: Design and performance of cognitive packet networks. Perform. Eval. **46**(2), 155–176 (2001)
62. Gelenbe, E., Lent, R., Xu, Z.: Measurement and performance of a cognitive packet network. Comput. Netw. **37**(6), 691–701 (2001)
63. Gelenbe, E., Lent, R., Xu, Z.: Towards networks with cognitive packets. In: Goto, K., Hasegawa, T., Takagi, H., Takahashi, Y. (eds.) Performance and QoS of Next Generation Networking, pp. 3–17. Springer, London (2001). https://doi.org/10.1007/978-1-4471-0705-7_1
64. Gelenbe, E., Liu, P.: QoS and routing in the cognitive packet network. In: Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, WoWMoM 2005, pp. 517–521. IEEE (2005)
65. Gelenbe, E., Loukas, G.: A self-aware approach to denial of service defence. Comput. Netw. **51**(5), 1299–1314 (2007)

66. Gelenbe, E., Mahmoodi, T.: Distributed energy-aware routing protocol. In: Gelenbe, E., Lent, R., Sakellari, G. (eds.) Computer and Information Sciences II, pp. 149–154. Springer, London (2011). https://doi.org/10.1007/978-1-4471-2155-8_18
67. Gelenbe, E., Nakip, M., Marek, D., Czachorski, T.: Diffusion analysis improves scalability of IoT networks to mitigate the massive access problem. In: IEEE MASCOTS 2021: 29th International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 1–6 (2021). https://zenodo.org/record/5501822#.YT3bri8itmA
68. Gelenbe, E., Ngai, E.C.H.: Adaptive QoS routing for significant events in wireless sensor networks. In: 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, MASS 2008, pp. 410–415. IEEE (2008)
69. Gelenbe, E., Ngai, E.C.: Adaptive random re-routing in sensor networks. In: Proceedings of the Annual Conference of ITA (ACITA 2008), 16–18 September, pp. 348–349 (2008)
70. Gelenbe, E., Ngai, E.C., Yadav, P.: Routing of high-priority packets in wireless sensor networks. In: IEEE Second International Conference on Computer and Network Technology. IEEE (2010)
71. Gelenbe, E., Nowak, M.P., Frohlich, P., Fiolka, J., Checinski, J.: Energy, QoS and security aware services at the edge. In: Gelenbe, E., et al. (Eds.) EuroCybersec 2021, CCIS 1596, pp. 102–117. Springer, Cham (2022)
72. Gelenbe, E., Pavloski, M.: Performance of a security control scheme for a health data exchange system. In: IEEE International Black Sea Conference on Communications and Networking, 26–29 May 2020, Virtual Conference (2020)
73. Gelenbe, E., Sakellari, G., D'arienzo, M.: Admission of QoS aware users in a smart network. ACM Trans. Auton. Adapt. Syst. (TAAS) **3**(1), 4 (2008)
74. Gelenbe, E., Sevcik, K.: Analysis of update synchronization for multiple copy data bases. IEEE Trans. Comput. **10**, 737–747 (1979)
75. Gelenbe, E., Sigman, K.: IoT traffic shaping and the massive access problem. In: ICC 2022, IEEE International Conference on Communications, Seoul, South Korea, 16–20 May 2022, pp. 1–6 (2022). https://zenodo.org/record/5918301#.YgaCP
76. Gelenbe, E., Wu, F.J.: Future research on cyber-physical emergency management systems. Future Internet **5**(3), 336–354 (2013)
77. Gelenbe, E., Xu, Z., Seref, E.: Cognitive packet networks. In: 11th IEEE International Conference on Conference Tools with Artificial Intelligence, Proceedings, pp. 47–54. Publisher IEEE (1999)
78. Gelenbe, E., Yin, Y.: Deep learning with random neural networks. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 1633–1638. IEEE (2016)
79. Gelenbe, E., Yin, Y.: Deep learning with dense random neural networks. In: Gruca, A., Czachórski, T., Harezlak, K., Kozielski, S., Piotrowska, A. (eds.) ICMMI 2017. AISC, vol. 659, pp. 3–18. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-67792-7_1
80. Gelenbe, S.E.: Cognitive packet network, uS Patent 6,804,201, 12 October 2004
81. Gorbil, G., Abdelrahman, O.H., Gelenbe, E.: Modeling and analysis of RRC-based signaling storms in 3G networks. IEEE Trans. Emerg. Top. Comput., 14 (2015). Special Issue on Emerging Topics in Cyber Security
82. Gorbil, G., Abdelrahman, O.H., Pavloski, M., Gelenbe, E.: Storms in mobile networks. arXiv preprint arXiv:1411.1280 (2014)

83. Gorbil, G., Abdelrahman, O.H., Pavloski, M., Gelenbe, E.: Modeling and analysis of RRC-based signalling storms in 3G networks. IEEE Trans. Emerg. Top. Comput. **4**(1), 113–127 (2016)
84. Gorbil, G., Gelenbe, E.: Opportunistic communications for emergency support systems. Procedia Comput. Sci. **5**, 39–47 (2011)
85. Gorbil, G., Gelenbe, E.: Resilience and security of opportunistic communications for emergency evacuation. In: Proceedings of the 7th ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks, pp. 115–124 (2012)
86. Jiang, H., Liu, F., Thulasiram, R.K., Gelenbe, E.: Guest editorial: special issue on green pervasive and ubiquitous systems. IEEE Syst. J. **11**(2), 806–812 (2017). https://doi.org/10.1109/JSYST.2017.2673218
87. Kadioglu, Y.M., Gelenbe, E.: Product-form solution for cascade networks with intermittent energy. IEEE Syst. J. **13**(1), 918–927 (2019)
88. Kehagias, D., Jankovic, M., Siavvas, M., Gelenbe, E.: Investigating the interaction between energy consumption, quality of service, reliability, security, and maintainability of computer systems and networks. SN Comput. Sci. **2**(1), 1–6 (2021)
89. Konar, D., Gelenbe, E., Bhandary, S., Sarma, A.D., Cangi, A.: Random quantum neural networks (RQNN) for noisy image recognition. CoRR abs/2203.01764 (2022)
90. Levi, A., Çaglayan, M.U., Koç, Ç.K.: Use of nested certificates for efficient, dynamic, and trust preserving public key infrastructure. ACM Trans. Inf. Syst. Secur. **7**(1), 21–59 (2004). https://doi.org/10.1145/984334.984336
91. Li, N., Hu, X., Ngai, E., Gelenbe, E.: Cooperative wireless edges with composite resource allocation in hierarchical networks. In: 2020 IEEE International Conference on E-Health Networking, Application & Services (HEALTHCOM), pp. 1–6 (2021). https://doi.org/10.1109/HEALTHCOM49281.2021.9398997
92. Liu, P., Gelenbe, E.: Recursive routing in the cognitive packet network. In: 3rd International Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities, TridentCom 2007, pp. 1–6. IEEE (2007)
93. Nakip, M., Gelenbe, E.: MIRAI botnet attack detection with auto-associative dense random neural networks. In: 2021 IEEE Global Communications Conference, vol. 2021, pp. 1–6. IEEE Communications Society (2021)
94. Nakip, M., Gelenbe, E.: Randomization of data generation times improves performance of predictive IoT networks. In: IEEE World Forum on Internet of Things (WF IoT), 14–21 July 2021, p. 5161 (2021). https://wfiot2021.iot.ieee.org
95. Nakip, M., Gelenbe, E.: Botnet attack detection with incremental online learning. In: In: Gelenbe, E., et al. (Eds.) EuroCybersec 2021, CCIS 1596, pp. 51–60. Springer, Cham (2022)
96. Nalin, M., et al.: The European cross-border health data exchange roadmap: case study in the Italian setting. J. Biomed. Inform. **94**, 103183 (2019)
97. Natsiavas, P., et al.: Developing an infrastructure for secure patient summary exchange in the EU context: lessons learned from the KONFIDO project. Health Inform. J. **27**(2) (2021). 14604582211021460
98. Natsiavas, P., et al.: Comprehensive user requirements engineering methodology for secure and interoperable health data exchange. BMC Med. Inform. Decis. Mak. **18**(1), 1–16 (2018)
99. Ngai, E.C., Gelenbe, E., Humber, G.: Information-aware traffic reduction for wireless sensor networks. In: IEEE 34th Conference on Local Computer Networks, LCN 2009, pp. 451–458. IEEE (2009)

100. Oke, G., Loukas, G., Gelenbe, E.: Detecting denial of service attacks with Bayesian classifiers and the random neural network. In: IEEE International Fuzzy Systems Conference, FUZZ-IEEE 2007, pp. 1–6. IEEE (2007)
101. Pavloski, M., Gelenbe, E.: Mitigating for signalling attacks in UMTS networks. In: Czachórski, T., Gelenbe, E., Lent, R. (eds.) Information Sciences and Systems 2014, pp. 159–165. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09465-6_17
102. Pavloski, M., Görbil, G., Gelenbe, E.: Bandwidth usage—based detection of signaling attacks. In: Abdelrahman, O.H., Gelenbe, E., Gorbil, G., Lent, R. (eds.) Information Sciences and Systems 2015. LNEE, vol. 363, pp. 105–114. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-22635-4_9
103. Pernici, B., Aiello, M., Vom Brocke, J., Donnellan, B., Gelenbe, E., Kretsis, M.: What is can do for environmental sustainability: a report from CAiSE? 11 panel on green and sustainable is. Commun. Assoc. Inf. Syst. **30**(1), 18 (2012)
104. Sakellari, G., Gelenbe, E.: Adaptive resilience of the cognitive packet network in the presence of network worms. In: Proceedings of the NATO Symposium on C3I for Crisis, Emergency and Consequence Management, pp. 11–12 (2009)
105. Sakellari, G., Hey, L., Gelenbe, E.: Adaptability and failure resilience of the cognitive packet network. In: DemoSession of the 27th IEEE Conference on Computer Communications (INFOCOM2008), Phoenix, Arizona, USA (2008)
106. Serrano, W., Gelenbe, E., Yin, Y.: The random neural network with deep learning clusters in smart search. Neurocomputing **396**, 394–405 (2020)
107. Siavvas, M., et al.: An empirical evaluation of the relationship between technical debt and software security. In: ICIST 2019 Proceedings, vol. 1, pp. 199–203 (2019)
108. Siavvas, M., Gelenbe, E.: Optimum checkpoints for programs with loops. Simul. Model. Pract. Theory **97** (2019)
109. Siavvas, M., Gelenbe, E.: Optimum interval for application-level checkpoints. In: 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), pp. 145–150. IEEE (2019)
110. Siavvas, M., Kehagias, D., Tzovaras, D., Gelenbe, E.: A hierarchical model for quantifying software security based on static analysis alerts and software metrics. Software Qual. J. **29**(2), 431–507 (2021). https://doi.org/10.1007/s11219-021-09555-0
111. Staffa, M., et al.: An openNCP-based solution for secure eHealth data exchange. J. Netw. Comput. Appl. **116**, 65–85 (2018)
112. Tugcu, T., Caglayan, M.U., Alagoz, F., Gelenbe, E.: New Trends in Computer Networks: 20th International Symposium on Computer and Information Sciences. World Scientific, September 2005. https://doi.org/10.1142/p415
113. Wang, L., Gelenbe, E.: Adaptive dispatching of tasks in the cloud. IEEE Trans. Cloud Comput. **6**(1), 33–45 (2018)
114. Yu, C., Ni, G., Chen, I., Gelenbe, E., Kuo, S.: Top-*k* query result completeness verification in tiered sensor networks. IEEE Trans. Inf. Forensics Secur. **9**(1), 109–124 (2014). https://doi.org/10.1109/TIFS.2013.2291326
115. Yu, C.M., Ni, G.K., Chen, Y., Gelenbe, E., Kuo, S.Y.: Top-k query result completeness verification in sensor networks. In: 2013 IEEE International Conference on Communications Workshops (ICC), pp. 1026–1030. IEEE (2013)

Application of a Human-Centric Approach in Security by Design for IoT Architecture Development

Violeta Vasileva(✉)

Artshare Ltd. – Intelligence, Technology and the Arts, Estarreja, Portugal
violeta.vasileva@artshare.com

Abstract. The aim of this paper is to show the need for a comprehensive approach to studying cybersecurity in order to get a focus on both, technological factors and the human aspect in development a user level of IoT Reference Architecture. The author presents a methodology for researching the capacity of users to manage their digital identity and operate in a safe and secure cyber-physical environment. For this purpose, an approach is proposed, that is based on a defined survey research targeting personnel and individual users operating in cyber-physical environment. Based on the analysed survey results and formed observations, the paper suggests solutions for improving the competences of users in cyberspace and increase their cyber awareness. As a practical implementation of the proposed methodology, the author presents a developed Cyber Awareness platform. Its main purpose is to provide information and resources on cybersecurity and IoT security, and to be the main point of knowledge access. The portal also provides opportunities to test users' cyber knowledge, participate in public survey related to cyber topics, share and exchange information, opinions and useful practices on cyber incidents and cyber knowledge.

Keywords: Cybersecurity · Human factor · IoT · Awareness · Survey

1 Introduction

1.1 Background

The rise of the Internet of Things (IoT) presents new opportunities for organizations to develop new technologies that will enable them to better manage their business and operations, create new products, improve daily work operations etc. However, implementing these new solutions can also lead to security issues. Even though there are a lot of studies related to Information Security, the research community seems to be little or no interested in the relationship between human factors and breaches.

There are broad range of issues related to European Cybersecurity and national experiences in the research in cybersecurity domain. A proper solution to respond to the complex phenomenon of cybersecurity is to implement human-centric approach in

E. Gelenbe et al. (Eds.): EuroCybersec 2021, CCIS 1596, pp. 13–22, 2022.
https://doi.org/10.1007/978-3-031-09357-9_2

system designs. Among the important topics is human factor (HF), as an object of vulnerability and cyber-attacks. This means that an important aspect in implementing the security by design approach are the cybersecurity aspects related to privacy by design [2] such as testing vulnerability of organization's cybersecurity of social engineering attacks, modelling HF in cybersecurity, developing HF Framework for cyber vulnerabilities investigations, as well as addressing the significance of cognitive user profiles for improving usability of computer system interface and others. These issues are relevant to the highest level of IoT Reference Architecture. Because the goal of the architecture is to improve security in IoT systems, therefore the human factor plays a role in adding security at the software application level as part of the user interfaces of the user domain.

Several studies have proved that half of data security breaches are caused by errors due to activities of ordinary users [1]. This contradicts the notion that hackers are the ones behind most breaches. Another key observation, from the study, is that human mistakes lead to more incidents than malicious actions do. That is why the attention should be focused on human factor activities. Moreover, in the current COVID-pandemic situation many users prefer working by the so-called "Work From Home" model. It requires more detailed research for cybersecurity in order to enhance user's cyber awareness and competence. According to ISACA's Covid-19 Study, about 90% of participants believe that an abrupt transition from the place of work to so called "home office" way of working would increase the risk for data privacy and cause problems [8].

A key aspect of the human factor is related to identity management, which is an important element of a cybersecurity system within organizations. With regard to the management of digital identity, key issues are security and privacy. As digital identity has become an increasingly popular attack vector and identity theft is widespread on the web, measures to identify and validate digital identities are crucial for network management and security in the public and private sectors.

1.2 Digital Identity as Object of Cybersecurity Vulnerabilities

Identity is the link that connects individuals to their community. It is a link between the individuals and the world in which they live. According to the APA[1] dictionary of Psychology [3], identity is "an individual's sense of self defined by (a) a set of physical, psychological, and interpersonal characteristics that is not wholly shared with any other person and (b) a range of affiliations (e.g., ethnicity) and social roles. Identity involves a sense of continuity, or the feeling that one is the same person today that one was yesterday or last year (despite physical or other changes). Such a sense is derived from one's body sensations; one's body image; and the feeling that one's memories, goals, values, expectations, and beliefs belong to the self. Also called personal identity."

The advent of digital technologies, the internet and social media have made possible the shift of the identity paradigm into a digital (cyber) context. The management of digital identity has many facets - technical, economic, social and cultural – and therefore it is complex to understand it and address it as a universal concept. Nevertheless, digital identity is essential for the further development of the Digital and Global Economy [7]. Many initiatives have been set to explore the necessity of unique digital identity

[1] American Psychological Association.

and it's adoption on a global scale. For example, The United Nations (UN) and World Bank ID4D initiatives aim to provide everyone on the planet with a legal identity by 2030. At the ID2020 summit in May 2016 in New York, the UN initiated discussions around digital identity, blockchain, cryptographic technologies, and its benefits for the underprivileged. During the summit, 400 experts shared best practices and ideas on how to provide universal identity to all.

The problem that exists is that in today's digital world it is a challenge to determine exactly what data such as type, quantity and quality are available in cyberspace. There is a huge amount of personal data on the Internet that is a "digital imprint" of a person and is linked to a "digital identity". Therefore, a key aspect of cyber security is the human factor.

As an open system, man communicates through the information environment and social networks. Through the physical environment the digital devices may turn into a means of striking, using its output to the person to broadcast massive amount of information or interfere with his everyday operations.

In addition, as a major challenge in IoT smart home infrastructure, is the security of devices. The main problem comes from the heterogeneous nature of IoT networks and large number of devices that differ on many criteria: the communication protocols that they use and data protocols [9]. Therefore, the security implementation will vary from device to device and finding a uniform solution is almost impossible [10]. Smart solutions contain a large amount of confidential data (e.g. personal photos, videos, microphones).

The notion of human factor within cyber-physical systems leads to one main observation in regard to user awareness within such systems.

In order to reach the above-mentioned aim of this paper, the issue has been explored, from a user's point of view, by applying a survey research methodology. It gave an opportunity to select key research indicators, related to cyber awareness and digital identity. They were explored and analysed, and then the evaluation criteria were defined. The study is aimed to help users get certain skills when they operate in the cyberspace. The survey was conducted and its results were compiled and examined. These results are used for the development of solutions that would enhance users' cyber awareness, digital competences and help them maintain high level cyberhygiene.

Based on the survey results, a human-centric perspective for a practical solution is applied by the development of a virtual online space - a website "Cyberawareness". It provides opportunities for active communication between users. They could submit information to the platform, as well as receive information from it. The virtual space of the platform has several functionalities. On one hand, it offers a tool for researching skills and behaviour of citizens, verifying their competencies in cyber awareness, and helps them manage their digital identity. On the other hand, it provides cyber awareness resources, access to various free cybersecurity verification tools, information resources, useful information and contacts, opportunities for sharing information related to cyber incidents, etc.

In this way, the platform could strengthen information sharing and improve cybersecurity competence of the workforce and citizens, especially in the sphere of the cybersecurity continuous education and the related services. According to ACM study group and the NICE framework (NIST Special Publication 800-181. Title. National Initiative for

Cybersecurity Education) it is necessary to pay attention on topics in the curricula like Organizational security (Security Operation and Personal Security), anonymizing data, Social Security (Customer service and technical support), Component Security (Procurement), Connection Security (Physical Interface and Connectors) due to the expansion of IoT-connected devices. It could be mentioned that the areas of utmost importance, like privacy by design, appeared to be present in less than 30% of the educational programmes. In addition, cybersecurity programmes in education should recognize the role of the human-centric factors, which provide basis to incorporate technology, software, organizational processes and users, and to study this as socio-technical and psychology system [4].

2 Approach for Improving Cyber Awareness Users When Working in Cyberspace and with IoT Devices' Interface

2.1 Research Method

The research method is based on the evaluation through analysis of collected data. The data which is collected provides information about the users, their occupation, experience and level of excellence. For the purpose of this study, the survey method was selected as the most appropriate one.

The survey examines and analyses the opinion of a wide range of participants that work in public and administrative structures. It provides an objective overview of the needs and potential areas for improvement in the organization of information infrastructures in those structures, as well as using IoT software applications.

The preparation of the survey is related to the following technology for work:

- Studying the general theory and technology of organizing and creating a survey;
- Exploring the possibilities for creating questionnaires or online forms with the relevant topics;
- Selection of indicators and definition of criteria for assessment of information needs;
- Development of a questionnaire;
- Organizing and conducting the survey;
- Processing and analysis of the results of the survey;
- Synthesis of the results, development of conclusions, recommendations and lessons for the users, administrators and developers;
- Applying the results of the survey to improve information security, defining them as information resources within the package of capabilities they need to implement.

Indicators for the Formation of the Criteria. As part of information security, indicators for the formation of criteria for cyber awareness assessment are related to capability building, as follows:

- Application of service-oriented architecture;
- Capability building for monitoring, detection and recognition of cyber threats in the IoT ecosystem environment;

- Capability building to ensure effective processing and sharing of information related to cyber incidents and threats.
- Personally Identifiable Information and Identity Management.

The needs for information security and services of the systems are planned and built-in accordance with their taxonomic grouping. It is a part of the comprehensive service-oriented, security by design and privacy by design approaches. The taxonomy is a hierarchical model consisting of a certain layer of services, including information security services.

Also, the capacity building for monitoring cyberspace is related to sharing data in real time through a subsystem for cyber incident detection.

In order to manage increased cyber risks and to improve cyber awareness, it is essential for users to counter detected threats, as well as to be capable and adaptable to neutralize new cyber threats generated by some innovative information technologies and especially those connected to IoT infrastructure and Personal Area Networks.

Evaluation Criteria. The definition of evaluation criteria is based on analysis of the proposed indicators for information security. It can be concluded that the basis for building the information environment should be a service-oriented architecture. That is why the evaluation criteria can be:

- applying a comprehensive service-oriented approach to build up information security;
- capability building for effective information sharing and processing among different organizational units;
- providing abilities to work in a group environment;
- getting specific abilities in order to observe, detect and recognize the environment in cyberspace in real time;
- assessment of the threat and risk level in regard to identity theft.

The set of indicators and defined criteria for evaluation of cyber awareness and information security are the basis that forms resources necessary to guarantee information security when users work in cyberspace.

A variant of such a consistent method for cyber awareness research, identity management, and proposing solution in order to enhance user competencies regarding cyberspace could be successfully applied, both at the national level for one country and in EU Cybersecurity Domain, as well.

The survey was conducted online between several different user groups. The “Forms” applications of the Microsoft Teams and Google Forms platforms have been used as implementation technology.

The analysis of the results of the survey is used in the development of a solution to increase cyber awareness.

3 Results of the Study

Given the dynamic state of information security vulnerabilities and the volume and complexity of existing threats, public organizations face a huge challenge defining and understanding human-related threats and risks. To help understand these challenges, cybersecurity experts and cyber users in public organizations from various institutions were interviewed through a survey. Based on the results of the survey, the key questions that all the users face are related to the lack of information about type of vulnerabilities, risk assessment, risk awareness, access to right information and support.

Interviews with the following three types of experts were conducted:

- users (of IoT)
- public sector workforce
- cyber security specialists and developers

Thirty-eight (38) people took part in the survey at the user level respondents. In the second survey, forty-three (43) respondents were interviewed, of which information security specialists were fifteen (15). Along with the specific answers to the questions, concrete proposals were made. The main one identified the need to provide opportunities for cyber awareness. These are opportunities to improve competencies of public sector users when working in cyberspace.

Some key observations are identified as follows:

- More than 50% of participants underwent security training within their organizations
- 78% of respondents say that in the corporations where they work there are established security policies. In a very small part of them these rules are related only to certain activities. Here the role of the leadership is positive and obvious.
- 72% of the respondents would accept when working in cyberspace to comply with the recommendations by observing their own user, monitor their device, knowing their correspondents, use virtual cards for your payments and more. However, 27% would not comply with such recommendations. Although a small percentage of dissenters, yet in a public administrative structure, such percentage is unacceptable.
- Over 60% of employees would like to be involved in training on the implementation of software capabilities to protect information and networks and especially IoT devices. The training can be done by external training structures, but also by the cyber security specialists.
- The results of the recommendations concern the exclusive administrator on information security. Highly recommended are end-user protection and awareness systems. They recommend having analysis systems in the information infrastructure, incident monitoring and reporting IoT vulnerabilities.
- More than 50% of the participants recommend the information infrastructure to have cyber threat systems and indicators, along with incident analysis, monitoring and reporting systems.
- More than 50% of the breakthroughs in the networks of corporate structures are due to the human factor of the organization itself and about 30% are caused by external factors and resources. This presupposes prevention in the work of the management with the users in the electronic environment and daily control over the users and the administrators in the IoT infrastructure. These conclusions also apply to network administrators for daily work with employees.

Based on the analysis of the results of the survey the following recommendations could be made towards the work of the users, managers and network and security administrators within public organizations.

Regarding Users and Managers in Institutions. The above data gathered from this group shows that justified cybersecurity concerns are present. This requires the leadership of institutions to develop measures and recommendations for safe work of their employees in the digital environment and enhance their cyber awareness.

- About cyber awareness level analysis – public sector organizations should create and use a cybersecurity policy during crises.
- About data analysis, smaller sized organizations with lower budgets and less employees are less prepared and less aware of cybersecurity risks. Therefore, more cybersecurity awareness is need.

Organizational measures are needed, including those related to identity management, such as:

Employees need to know that each account matches exactly a particular user and everyone must act responsibly and protect their data. Username information and passwords should not be provided of third parties, as well as in various digital platforms and social media. In case of suspicions about profile theft or compromised account, notify immediately the security administrator in the institution.

Password compromising is one of the main reasons for most cybercrime incidents. Quality management of passwords assume that each account is secured with a unique one access password. Passwords must be sufficient at the same time long and complex enough to be composed of different characters and symbols. Passwords should include words, names or anything that is easy to associate with their owners.

Improper password management can lead to significant risks of theft and irreversible loss of information, leakage of sensitive data, and breakthrough in information systems.

Passwords are strictly personal and on no occasion and under no circumstances should not be shared - be it sent by e-mail, recorded on paper, communicated by telephone, fax or other insecure or easy to read format or channel, and under no circumstances should be entered in electronic surveys. Passwords must not be saved in a file on a workstation, server, or mobile device in unencrypted view.

Compliance with security policies by users will ensure a relatively safe working environment in cyberspace. Administrators must also commit to compliance with network and security policies, as well as the management of the institution's information infrastructure.

Since the vision of IoT is to connect as many smart devices as possible, it is important that IoT users have all data available and at all times. However, data is not the only component used in IoT. Devices and services must also be available in a timely manner when needed to achieve IoT expectations. Denial of service (DDoS) attack is an example of affecting resource availability.

Information systems and services that we use are becoming safer and "more secure" to hack, but in fact people remain the weakest part of the cybersecurity system. Their mistakes can compromise the whole system. Therefore, it is important how to promote cyber hygiene and consumer behaviour through cyber awareness, education and training which corresponds to advances in psychology, state of the art technologies and security.

The human factor vulnerability is a major target of social engineering attacks, which completely circumvent all technical protection measures taken. Social engineering is method for unauthorized acquisition of information resources and/or user rights without the use of technical means. Social engineering uses mainly psychological methods, namely a person's tendency to trust. Social attacks engineering take place on two levels:

- Physical level are offices, telephones, trashcans, business mail.
- The social engineer can simply enter the workplace, posing as a maintenance person, and to get a custom username and password.

This psychological approach uses well-established methods for persuasion: presenting to someone else, conformism, reference to authoritative figure, distraction or just friendly attitude. The most common and easy way to get a third party with username and password is by receiving it directly from the user through various methods of persuasion, deception, involuntary sharing, misleading in order to achieve financial benefits, etc. Social engineering is the preferred method to launch an attack on a system because in case of carelessness on the part of the user the attacker can easily obtain the necessary information.

The human factor is in conjunction with the implementation of security policies. It is important to note the need for the rules for security policies of IoT infrastructure devices to be developed according to the participants in the overall IoT ecosystem. They could be summarized as security policies for developers and service providers (in the processes of infrastructure development, implementation and integration), as well as security policies for end users of IoT devices and applications [5, 6]. Security is not only about technology, but also about the people. That is why it should be accepted that the human factor in cybersecurity cannot be ignored. Therefore, it is necessary to take actions and care of people, such as: education - to make users aware; establishing proper security policies; to constantly study one's own mistakes and weaknesses.

Developing the Project Portal "Cyber Awareness". In line with the observations based on survey results, the author presents a methodology for enhancing the knowledge of users to work in a secure environment and to refine and improve their cyber awareness and IoT security skills. As a practical implementation of the proposed methodology, the author presents a developed Cyber Awareness portal. Its main purpose is to provide information and resources on cybersecurity and IoT security, and to be the main point of knowledge access. The portal also provides opportunities to test your cyber knowledge, participate in public survey related to cyber topics, share and exchange information, opinions and useful practices on cyber incidents and knowledge.

4 Conclusion

IoT technologies became everyday commodities, peculating both our social and work environments. The complexity, variety, and frequency of cyber incidents have increased significantly over the last decade. Understanding cybersecurity risk requires a certain level of discipline and cautious mentality, which requires increasing cyber awareness levels among regular users of IoT.

The developed portal is a good practice which can be further improved in a collaborative platform that enables regular editing and enrichment of the content with additional up-to-date information, setting up a forum and discussion structure, as well as adding tools. The proposed approach for improving cyber awareness of people has an interdisciplinary effect, as it could be successfully applied in other cyber domains. Human vulnerabilities need to be identified and managed before they lead to an actual security breach. For that reason, we need a common research perspective to study cybersecurity that focuses on the interaction of technology and software development, concepts and architectures, organizational processes improvement and human performance. This means to apply a human–centred approach, which provides a comprehensive foundation to analyse cybersecurity as a socio-technical system that cover diverse aspects such as psychological, cultural, technology and software development.

References

1. 2018 IT Risks Report: Netwrix Corporation. netwrix.com
2. Annual Report 2020: European Data Protection Supervisor PDF ISBN 978-92-9242-617-0 ISSN 1830-9585. https://doi.org/10.2804/205036QT-AA-21-001-EN-N. https://edps.europa.eu/system/files/2021-04/2021-04-19-annual-report-2020_EN.pdf
3. APA dictionary of Psychology. https://dictionary.apa.org/
4. Borka Jerman Blažič (March 31st 2021): Cybersecurity Skills in EU: New Educational Concept for Closing the Missing Workforce Gap. IntechOpen. https://doi.org/10.5772/intechopen.97094. https://www.intechopen.com/online-first/75922
5. Department for Digital, Culture, Media & Sport UK: Code of Practice for Consumer IoT Security (2018). https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/971440/Code_of_Practice_for_Consumer_IoT_Security_October_2018_V2.pdf
6. Department of Home Affairs Australia: Code of Practice "Securing the Internet of Things for Consumers" (2020). https://www.homeaffairs.gov.au/reports-and-pubs/files/code-of-practice.pdf
7. Digital Identity Management Enabling Innovation and Trust in the Internet Economy – OECD (2011). https://www.oecd.org/sti/ieconomy/49338380.pdf
8. ISACA's COVID-19 Study (2020). www.isaca.org/covid19study
9. Larsson, M., Lipkin, L.: Exploring the viability of intrusion detection in a centralized smart hub (2018). https://odr.chalmers.se/handle/20.500.12380/256483
10. Rehman, S., Manickam, S.: A study of smart home environment and its security threats (2016). https://www.researchgate.net/publication/303089918_A_Study_of_Smart_Home_Environment_and_it%27s_Security_Threats

An Empirical Evaluation of the Usefulness of Word Embedding Techniques in Deep Learning-Based Vulnerability Prediction

Ilias Kalouptsoglou[1,2](✉), Miltiadis Siavvas[1], Dionysios Kehagias[1], Alexandros Chatzigeorgiou[2], and Apostolos Ampatzoglou[2]

[1] Centre for Research and Technology Hellas, Thessaloniki, Greece
{iliaskaloup,siavvasm,diok}@iti.gr
[2] University of Macedonia, Thessaloniki, Greece
achata.ampatzoglou@uom.edu.gr

Abstract. Software security is a critical consideration for software development companies that want to provide their customers with high-quality and dependable software. The automated detection of software vulnerabilities is a critical aspect in software security. Vulnerability prediction is a mechanism that enables the detection and mitigation of software vulnerabilities early enough in the development cycle. Recently the scientific community has dedicated a lot of effort on the design of Deep learning models based on text mining techniques. Initially, Bag-of-Words was the most promising method but recently more complex models have been proposed focusing on the sequences of instructions in the source code. Recent research endeavors have started utilizing word embedding vectors, which are widely used in text classification tasks like semantic analysis, for representing the words (i.e., code instructions) in vector format. These vectors could be trained either jointly with the other layers of the neural network, or they can be pre-trained using popular algorithms like word2vec and fast-text. In this paper, we empirically examine whether the utilization of word embedding vectors that are pre-trained separately from the vulnerability predictor could lead to more accurate vulnerability prediction models. For the purposes of the present study, a popular vulnerability dataset maintained by NIST was utilized. The results of the analysis suggest that pre-training the embedding vectors separately from the neural network leads to better vulnerability predictors with respect to their effectiveness and performance.

Keywords: Software security · Vulnerability prediction · Deep learning · Natural language processing · Word embedding vectors

1 Introduction

Vulnerability Prediction (VP) techniques aim to identify the software components that are more likely to contain vulnerabilities. Vulnerability prediction

E. Gelenbe et al. (Eds.): EuroCybersec 2021, CCIS 1596, pp. 23–37, 2022.
https://doi.org/10.1007/978-3-031-09357-9_3

models (VPMs) are typically built using machine learning (ML) techniques that use software attributes as input to differentiate between vulnerable and clean (or neutral) software components. Several VPMs have been proposed over the years, each of which uses different software factors as inputs to predict the presence of vulnerable components [1]. Text mining-based techniques have been found to be the most reliable, according to the bibliography [2] and have attracted the most of the recent research interest [3–10]. The first attempts in the field of vulnerability prediction using text mining, have focused on the concept of Bag-of-Words (BoW) as a method for predicting software vulnerabilities using the text terms and their respective appearance frequencies in the source code. Recently, researchers have shifted their focus from simple BoW to more complex approaches, investigating whether more complex textual patterns in the source code could lead to more accurate vulnerability prediction. In particular, the authors in [4,5,8] transformed the source code into sequences of word tokens and trained deep neural networks capable of learning sequences of data

When using sequences of tokens to identify software components with vulnerabilities, vulnerability prediction has a lot in common with text classification tasks such as sentiment analysis [11]. The word embedding vectors are commonly used in the field of text classification. Word embedding is a term used to describe the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word in such a way that words that are close in the vector space are expected to have similar meanings. Most of the studies about VP make use of the word embeddings [4–6,12].

Tokens can be embedded into vectors in a variety of ways. One option is to use an embedding layer that is jointly trained with the vulnerability prediction task [13]. Another method is to use an external word embedding tool, such as word2vec [14], to generate vector representations of each token. One can also use the vectors that are already generated from these tools (e.g. word2vec, Glove [15], Fast-text [16]) based on natural language documents of billions of words. Finally, there is also the option to produce custom embedding representations.

The purpose of this study is to emerge the worth of the sophisticated embedding algorithms (e.g., word2vec, fast-text) in text mining-based vulnerability prediction showing their contribution to the effectiveness and the efficiency of the VPMs and to compare them with the use of a trainable embedding layer that updates its values during the training of the VP classifier. A dataset has been collected and an experimental analysis has been conducted by comparing the use of a simple embedding layer with the utilization of word2vec and fast-text algorithms. We also compare the word2vec and fast-text algorithms with each other. Finally, we compare our best model with a state-of-the-art model.

The remainder of the paper is organized as follows. Section 2 discusses related work regarding the utilization of word embeddings in the field of vulnerability prediction. Section 3 provides the theoretical background in order to familiarize the reader with the main concepts of the present work. Section 4 discusses the methodology that we followed, while Sect. 5 presents the results of our analysis. Finally, Sect. 6 wraps up the paper and discusses future research directions.

2 Related Work

Vulnerability prediction using text mining is very popular and has demonstrated promising results in the related literature [2,4,9,10]. Initial research endeavors focused on the concept of BoW (i.e., occurrences of tokens) [2,9]. Recent attempts focus on predicting the existence of vulnerabilities through learning more complex patterns from the source code. They consider the software components as sequences of tokens and train deep learning models capable of learning sequences, such as the Recurrent Neural Networks (RNNs) [4,5]. The challenging part of these recent studies is to add syntactic and semantic meaning to the sequences of code tokens. Word embeddings are one of the most promising solutions.

The word embedding vectors have evolved into an integral part of the text classification tasks since Mikolov et al. [17] proposed two architectures for learning distributed representations of words. The authors in [18], conducted a comparative study between different ML algorithms including fast-text, Glove and word2vec, while in [19] a deep learning method is proposed utilizing the semantic knowledge provided by the word embeddings.

Word embeddings have already been used in the field of text mining-based vulnerability prediction. Dam et al. [8] mapped every code token with an index of their vocabulary and then they constructed an embedding matrix which contained a unique vector representation for every token of the vocabulary in the position that corresponds to the vocabulary index. In other words, the embedding matrix worked as a look-up table.

The authors in [5,6] used the word2vec tool to generate embedding vectors for their vocabulary, while Zhou et al. [4] used the pre-trained word2vec vectors. Russel et al. in [12] created a vulnerability detection tool based on deep learning and capable of interpreting lexical source code. They conducted a comparative study between simple source code embedding using Bag-of-Words and more advanced code representations learned automatically by deep learning models inside the embedding layer. Fang et al. [20] proposed the fastEmbed model which is an extension of the fast-text algorithm. This way they developed a model for predicting the exploitability of software vulnerabilities on imbalanced datasets by understanding key features of vulnerability-related text.

To this end, it is quite clear that a lot of studies make use of word embedding vectors as a representative format for the source code's tokens (i.e., words). There are papers that refer the use of simple vector representations just in order to replace the text features [8], other papers that use the BoW methodology to represent the text in the source code [9], other studies that utilize the pre-trained embedding vectors produced by the pre-trained word2vec model [4], but most of them choose to encode the code tokens into embedding vectors trained on their own data [5,6]. However, to the best of our knowledge, there is no study examining the difference between the internal embeddings that are trained in the embedding layer together with the classifier, and the external embeddings that are trained alone prior to the model's training. The former are part of the supervised learning of the model and update their weights through the Backpropagation process [21], while the latter are trained once, using an advanced unsupervised algorithm, and then they can be saved for future use. Moreover, there is a need for an experimental analysis examining the improvement in terms of accuracy and performance that these word embeddings provide to the DL-based vulnerability predictors. In the present work, we attempt to address these open issues through an empirical analysis on a popular dataset. Furthermore, the present paper includes a comparison between two popular types of word embedding tools (i.e., word2vec, fast-text) as well as a comparison with a state-of-the-art BoW model.

3 Theoretical Background

3.1 Vulnerability Prediction Based on Text-Mining

Vulnerability Prediction purpose is to identify software hotspots that are more likely to contain software vulnerabilities. These hotspots are actually parts of the source code that require more attention by the software developers and engineers from a security viewpoint. When the VPMs are based on text-mining they are trained on datasets constructed by the words (i.e., tokens) that appear in the source code. BoW constitutes the simplest text-mining method. In BoW, the code is divided into text tokens, each one of which is accompanied by the number of its occurrences in the source code. So each word corresponds to a feature, and the frequency of that feature in a component adds up to the value of that feature for that component. Aside from BoW, text-mining includes the process of converting the source code into a list of token sequences for use as input to Deep Learning (DL) models capable of parsing sequential data (e.g., recurrent neural networks). The sequences of tokens constitute the input of the DL models that, during the training phase, try to capture the syntactic information included in the source code, and in the execution phase to predict the existence of vulnerabilities in the software components. Text-mining also uses Natural Language Processing (NLP) methodologies such as word2vec pre-trained embedding vectors to extract semantic information from tokens.

3.2 Word Embedding Vectors

Word embedding methods use a corpus of text to learn a real-valued vector representation for a predefined fixed-sized vocabulary [17]. The learning process is either collaborative with the neural network model on a task, or unsupervised, using document statistics. An embedding layer is a word embedding learned in conjunction with a neural network model on a specific natural language processing task, such as document classification. It necessitates cleaning and preparing the document text so that each word can be one-hot encoded. The model specifies the size of the vector space. The vectors are seeded with small random numbers. The embedding layer is used at the front end of a neural network and is fitted in a supervised manner using the Backpropagation algorithm. However, it can be selected to be non-trainable. In this case, it has to be seeded with a pre-trained embedding matrix which has been trained using an external algorithm.

Mikolov et al. [17] proposed two model architectures for computing continuous vector representations of words. They showed that these representations were able to capture syntactic and semantic word similarities. Both architectures are neural network-based ones for learning the underlying word representations for every word. The first proposed model, called Continuous Bag-of-Words Model (CBOW), tends to find the probability of a word occurring in a context. Thus, it generalizes over all the different contexts in which a word can be used. The second architecture, called continuous skip-gram model, instead of predicting the current word based on context, attempts to maximize classification of a word based on another word in the same sentence. To be more specific, every current word is fed into a log-linear classifier with a continuous projection layer, which predicts words within a certain range before and after the current word.

Two of the most popular algorithms that can generate embedding vectors are the word2vec[1] and fast-text[2] models. Both of them are based on the two aforementioned architectures (i.e., CBOW, skip-gram). The difference between these tools lies in the fact that the word2vec considers each individual word to be the smallest unit for which a vector representation must be found, whereas fast-text considers a word to be formed by n-grams of character.

4 Methodology

4.1 Dataset

As part of the current work, we created several VPMs for two widely-used programming languages, C and C++ combined. We used a vulnerability dataset derived from two National Institute of Standards and Technology (NIST) data sources: the National Vulnerability Database (NVD[3] and the Software Assurance Reference Dataset (SARD)[4]. This dataset contains 7651 class files, 3438 of which are classified as vulnerable and the remaining 4213 as clean. The dataset has been presented by Li et al. [5].

4.2 Pre-processing

Before the construction of vulnerability prediction models, appropriate pre-processing is required in order to bring the dataset in a form appropriate to be used by the investigated techniques. To this end, we gathered the source code files written in the C and C++ programming languages and used a variety of pre-processing techniques to convert the datasets into a series of words-tokens. All comments, as well as the header/import instructions that declare the use

[1] https://radimrehurek.com/gensim/models/word2vec.html.
[2] https://radimrehurek.com/gensim/models/fasttext.html.
[3] https://nvd.nist.gov/.
[4] https://samate.nist.gov/SRD/index.php.

of specific libraries in the class, were removed from the dataset. Subsequently, we removed the code-specific constants (i.e., numbers, literals, etc.), in order to make the produced sequences more generalizable. In particular, the numeric values (i.e., integers, floats, etc.) were then replaced by a unique identifier "numId$", while the string values and characters were replaced by a different unique identifier "strId$". All blank lines are also removed, and the text is finally transformed into a list of code tokens (i.e., new, char, strlen, etc.) in the order they appear in the source code. After data cleansing, these produced tokens are replaced by a unique integer (integer encoding process[5]) and these integers are mapped to one-hot vectors (one-hot encoding[6]).

4.3 Word Embedding Vectors Training

In this study, in order to embed the text representations to numerical vectors different from the one-hot vectors, the word2vec and fast-text tools were utilized. The two models provide both the CBOW and skip-gram architecture. We train these models with our dataset. Each software component constitutes a sequence of tokens and all the sequences of the dataset are used as the corpus for the training of the word2vec and fast-test models. These algorithms learn the syntactic and semantic relations between the code tokens and place them at the vector space. After training these embedding vectors for the words of the vocabulary then one can save them for future use, saving time of the training process. For the training of the embedding vectors, the parameters that were selected after tuning are listed in Table 1.

Table 1. The selected parameters for the training of word2vec and fast-text embedding vectors.

Parameters	Word2vec	Fast-text
Size	300	300
Window	40	40
min_count	1	1
Epochs	1	2

In Table 1, the parameter "size" is the dimension of the embedding vectors, the "window" refers to the maximum distance between a target word and words surrounding the target word while the term "min_count" refers to the minimum count of words to consider during training. The algorithm ignores the words with

[5] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html.

[6] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html.

occurrence less than the "min_count". The parameter epochs is just the number of iterations that the model parses the data.

In Fig. 1, there is a depiction of word2vec vectors trained at the dataset used in the present study, generated by the t-distributed stochastic neighbor embedding (TSNE) algorithm [22]. Vectors that are in close proximity in the depicted figure, correspond to words that are in close proximity in the actual source code. For instance, in Fig. 1 we can see that the tokens "for" and "i" which actually are used together in a lot of circumstances, are indeed placed one next to the other. The same applies also for the tokens "free" and "malloc", which is another very representative example.

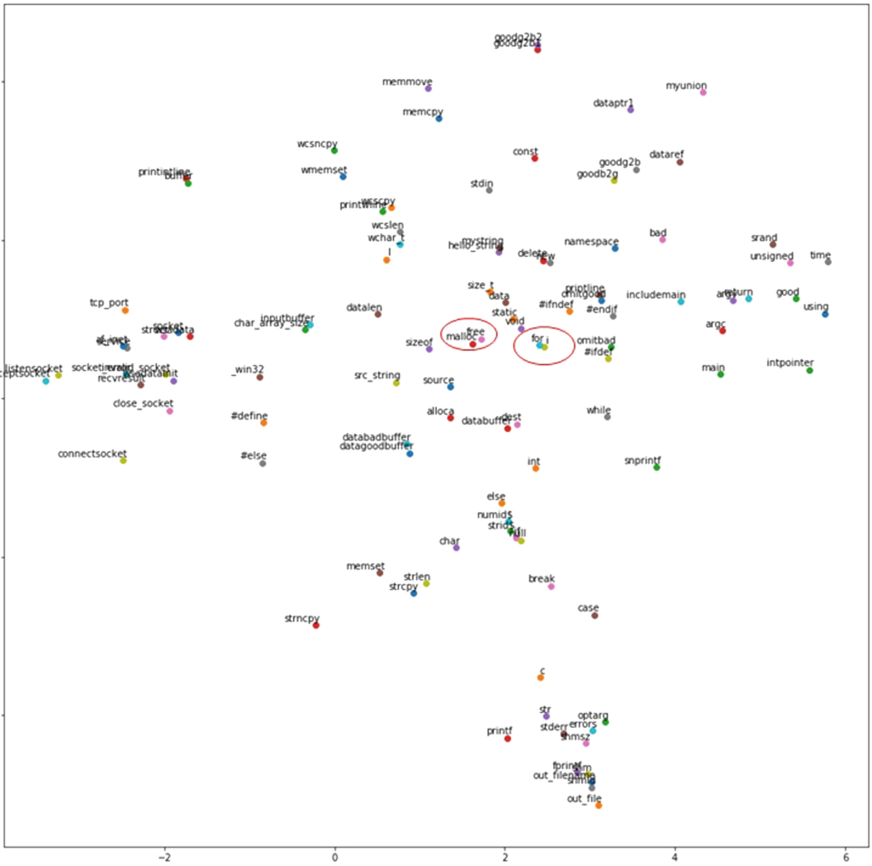


Fig. 1. The word2vec embedding vectors placed in the vector space by the TSNE algorithm.

4.4 Model Selection

In this analysis, various DL algorithms are used to create models that can distinguish between vulnerable and neutral source code files. As the input to the models consists of sequential data (i.e., series of tokens) we chose DL algorithms capable of handling sequences. The RNNs are the most suitable ones as language models [23]. Convolutional Neural Networks (CNNs) are used on code classification tasks, as well [4,24]. Regarding the RNNs, there are several improved versions such as the Long-Short Term Memory networks (LSTMs) [25], the Gate Recurrent Units (GRUs) [26] and the Bidirectional LSTMs (BiLSTMs) [27], which can solve the vanishing gradient problem [28] that the original RNNs face. The hyper-parameters chosen for our RNN and CNN models are presented in Table 2. Their values were selected after consecutive tuning and re-evaluation.

Table 2. The selected hyper-parameters of the models.

Hyper-parameter name	Value for RNNs	Value for CNN
Number of layers	3 (Embedding-Recurrent-Dense)	3 (Embedding-Conv-Dense)
Number of hidden layers	1 (LSTM/GRU/BiLSTM)	1 (1D CNN)
Embedding size	300	300
Number of hidden units	300	128 filters
Kernel size	–	5
Weight initialization technique	Glorot Uniform (Xavier)	Glorot Uniform (Xavier)
Learning rate	0.01	0.01
Gradient descent optimizer	Adam	Adam
Batch size	64	64
Activation function	Relu	Relu
Output activation function	Sigmoid	Sigmoid
Loss function	Binnary Cross-entropy	Binnary Cross-entropy
Over-fitting prevention	Dropout = 0.3	–
Maximum epochs	100	100
Early stopping patience	10	10

4.5 Evaluation Metrics

Several evaluation metrics are available in the literature and are commonly used to assess the predictive effectiveness of the ML models. In the vulnerability prediction case, a special emphasis is placed on the Recall of the produced models, because the higher the Recall of the model is, the more real vulnerabilities it predicts. Apart from the capability of the produced models to identify the great majority of vulnerable files contained in a software project, the volume of the produced False Positives (FP) (i.e., clean files marked as vulnerable) is important to consider because it is known to affect the models' utilization in practice. The number of FP is closely related to the amount of manual effort required by developers to identify files that contain actual vulnerabilities. The lower the number of FP is, the higher the precision of the model. This fact emphasizes the significance of the f1-score, which represents the balance of precision and recall. However, because identifying vulnerable files at the expense of producing FP is more important in VP, we chose f2-score as our evaluation metric. The f2-score is a weighted average of precision and recall, with recall being more important than precision. It is equal to:

$$F_2 = \frac{5 \times precision \times recall}{4 \times precision + recall}$$

5 Results and Discussion

In this section, we present the results of our analysis and discuss the outcome of the experiments. Table 3 reports the evaluation results of the DL models that were built based on the sequences of tokens in the source code. This table sums up the results regarding the f2-score for all the RNN variations and CNN using word2vec or fast-text embeddings in contrast with the joint training of the embeddings with the neural network's training. During the evaluation process, the ten-fold cross-validation process was employed eliminating the possibility of biased results.

Table 3. The f2-score of all the utilized methods after 10 fold cross-validation.

Model	S.E.L.	W2V 0	W2V 1	FastText 0	FastText 1
LSTM	77.98	85.11	88.38	84.92	88.66
BiLSTM	80.28	85.86	88.01	82.33	86.04
GRU	72.22	89.15	87.94	84.28	89.10
CNN	81.46	86.36	89.43	86.36	84.54
Average	77.99	86.62	88.44	84.47	87.09

S.E.L. = Simple Embedding Layer.
W2V: Word2vec.
0: CBOW.
1: Skip-gram.

From Table 3, we can see that the use of sophisticated word embeddings trained prior to the deep learning model is beneficial at each model case. The average f2-score of the four models when using an algorithm for the generation of the word embedding vector is significantly bigger. Furthermore, it is clear that the skip-gram model is better than the CBOW in our dataset as it achieves greater average f2-score both at the word2vec and the fast-text case. Similarly, we notice that the word2vec method provides better f2-score, both at the CBOW and the skip-gram variation, compared with the fast-text embeddings. All the aforementioned foundings lead to the conclusion that the skip-gram variation of the word2vec embeddings is the best choice for embedding the tokens of the source code of our dataset before giving them as input to the sequential deep learning model. An 11% increase in terms of f2-score when using word2vec embeddings compared with the trainable embedding layer is a significant improvement and indicates to the initial hypothesis that these sophisticated models are capable of capturing semantic and syntactic relationships between the words of the source code.

Furthermore, the training of the embedding vectors outside from the embedding layer (i.e., non-trainable embedding layer) is beneficial not only in accuracy but also in terms of performance. The training time of the DL models has decreased significantly. Table 4 sums up the results about the training time.

Table 4. The training time in milliseconds (ms) both in case of trainable embedding layer and in case of sophisticated embeddings trained independently of the neural network.

Model	S.E.L.	W2V 1
LSTM	13078	9090
BiLSTM	22596	18011
GRU	12025	8330
CNN	9774	4276

From Table 4, it is clear that the training times when having ready the embedding vectors are by far smaller compared with the case of joint training along with the rest layers. Another interesting note derived from Table 3 and 4 is the fact that the CNN model is more accurate than the RNNs and much faster as well.

Finally, another interesting question would be to examine whether the adoption of word embedding vectors lead to better vulnerability prediction models compared to the traditional (and simpler) BoW approach. For this purpose, we compare our best model that utilizes the word embedding concept to the best model that uses BoW and is trained and evaluated on the same dataset. In particular, in Table 5, we present the results of the comparison between the state-of-the-art BoW method, versus the CNN model with skip-gram word2vec vectors, which was found to be the best model in our previous analysis. In the case of BoW, we chose Random Forest (composed of 100 trees) as a classifier, based on bibliography [2,9]. From Table 5, it is observed that the f2-score is greater in the case of using sophisticated embeddings. Actually, these word2vec vectors can be used only at token series models (i.e., CNN, RNN) and not in BoW, constituting a major drawback of the method.

Table 5. BoW versus CNN that uses the skip-gram word2vec representations.

Model	Accuracy	Precision	Recall	F2-score
BoW	88.69	90.40	85.80	86.66
CNN-W2V 1	88.25	86.21	90.31	89.43

6 Conclusion and Future Work

In this paper, we investigated the usefulness of the numerical representations of the source code words, with the aim of predicting vulnerabilities. We focused on examining whether the utilization of sophisticated (i.e., external) embedding vectors is beneficial in contrast with the training of the embedding vectors jointly with the vulnerability predictor. Moreover, a comparison between the CBOW and the continuous skip-gram architectures took place as well as a comparison between the word2vec and fast-text algorithms.

We showed that either the word2vec or fast-text methodologies provide better results than the trainable embedding layer which is trained along with the rest layers of the neural network. These vector representations seem able to capture semantic and syntactic relations between the words in the code and so they can be proved beneficial when training models on sequences of code tokens. The word2vec method proved to be superior to fast-text when applied in our dataset. Furthermore, the skip-gram model demonstrated better scores compared with the CBOW, both in cases of word2vec and fast-text. Another important advantage of these sophisticated vectors is the time reduction during the model training, as there is no need to train the embedding layer again. Last but not least, the CNN with trained word2vec embeddings, which appeared to be our best model, demonstrates higher f2-score than the BoW model.

There are several potential directions for future work. First of all, the present study was based on a dataset containing exclusively C/C++ code. We intend to replicate our study using software products written in other programming languages (e.g., Java, Python, etc.) to investigate the generalizability of the produced results. Furthermore, we aim to replicate our study by embedding the text features in a higher level of granularity (e.g., line or function level).

Acknowledgements. This work is partially funded by the European Union's Horizon 2020 Research and Innovation Programme through IoTAC project under Grant Agreement No. 952684.

Appendix

In the Appendix Section we provide some extra Tables of the results produced by our study, including values for accuracy, precision and recall aside from the f2-score.

Simple Embedding Layer:

Model	Accuracy	Precision	Recall	F2-score
LSTM	75.61	74.69	79.60	77.98
BiLSTM	77.05	76.06	82.73	80.28
GRU	73.06	72.60	72.34	72.22
CNN	85.15	88.43	79.96	81.46

Word2vec Embeddings - CBOW:

Model	Accuracy	Precision	Recall	F2-score
LSTM	78.76	74.23	88.90	85.11
BiLSTM	80.62	75.79	88.99	58.86
GRU	84.55	79.52	92.02	89.15
CNN	86.60	86.16	86.52	86.39

Word2vec Embeddings - Skip-Gram:

Model	Accuracy	Precision	Recall	F2-score
LSTM	84.07	79.35	91.14	88.38
BiLSTM	84.51	80.26	90.31	88.01
GRU	83.44	78.51	90.74	87.94
CNN	88.25	86.21	90.31	89.43

Fast-Text Embeddings - CBOW:

Model	Accuracy	Precision	Recall	F2-score
LSTM	79.79	75.11	88.07	84.92
BiLSTM	76.20	71.15	85.79	82.33
GRU	77.58	72.71	88.53	84.28
CNN	86.08	84.98	86.82	86.36

Fast-Text Embeddings - Skip-Gram:

Model	Accuracy	Precision	Recall	F2-score
LSTM	83.54	78.23	91.79	88.66
BiLSTM	79.20	73.49	90.01	86.04
GRU	82.38	76.57	93.11	89.10
CNN	85.15	84.87	84.58	84.57

References

1. Siavvas, M., Gelenbe, E., Kehagias, D., Tzovaras, D.: Static analysis-based approaches for secure software development. In: Gelenbe, E., et al. (eds.) Euro-CYBERSEC 2018. CCIS, vol. 821, pp. 142–157. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95189-8_13
2. Walden, J., Stuckman, J., Scandariato, R.: Predicting vulnerable components: software metrics vs text mining. In: 2014 IEEE 25th International Symposium on Software Reliability Engineering. IEEE (2014)
3. Chakraborty, S., Krishna, R., Ding, Y., Ray, B.: Deep learning based vulnerability detection: are we there yet. IEEE Trans. Softw. Eng. (2021)
4. Zhou, Y., Liu, S., Siow, J., Du, X., Liu, Y.: Devign: effective vulnerability identification by learning comprehensive program semantics via graph neural networks. arXiv preprint arXiv:1909.03496 (2019)
5. Li, Z., et al.: Vuldeepecker: a deep learning-based system for vulnerability detection. arXiv preprint arXiv:1801.01681 (2018)
6. Cao, S., Sun, X., Bo, L., Wei, Y., Li, B.: BGNN4VD: constructing bidirectional graph neural-network for vulnerability detection. Inf. Softw. Technol. **136**, 106576 (2021)
7. Pang, Y., Xue, X., Wang, H.: Predicting vulnerable software components through deep neural network. In: Proceedings of the 2017 International Conference on Deep Learning Technologies, pp. 6–10 (2017)

8. Dam, H.K., Tran, T., Pham, T.T.M., Ng, S.W., Grundy, J., Ghose, A.: Automatic feature learning for predicting vulnerable software components. IEEE Trans. Softw. Eng. **47**, 67–85 (2018)
9. Scandariato, R., Walden, J., Hovsepyan, A., Joosen, W.: Predicting vulnerable software components via text mining. IEEE Trans. Softw. Eng. **40**, 993–1006 (2014)
10. Hovsepyan, A., Scandariato, R., Joosen, W., Walden, J.: Software vulnerability prediction using text analysis techniques. In: Proceedings of the 4th International Workshop on Security Measurements and Metrics, pp. 7–10 (2012)
11. Medhat, W., Hassan, A., Korashy, H.: Sentiment analysis algorithms and applications: a survey. Ain Shams Eng. J. **5**(4), 1093–1113 (2014)
12. Russell, R., et al.: Automated vulnerability detection in source code using deep representation learning. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 757–762. IEEE (2018)
13. Turian, J., Ratinov, L., Bengio, Y.: Word representations: a simple and general method for semi-supervised learning. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 384–394 (2010)
14. Rong, X.: word2vec parameter learning explained. arXiv (2014)
15. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
16. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
17. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
18. Stein, R.A., Jaques, P.A., Valiati, J.F.: An analysis of hierarchical text classification using word embeddings. Inf. Sci. **471**, 216–232 (2019)
19. Ma, Y., Peng, H., Cambria, E.: Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
20. Fang, Y., Liu, Y., Huang, C., Liu, L.: Fastembed: Predicting vulnerability exploitation possibility based on ensemble machine learning algorithm. Plos One **15**, e0228439 (2020)
21. Goldberg, Y.: Neural network methods for natural language processing. Synth. Lect. Hum. Lang. Technol. **10**(1), 1–309 (2017)
22. Van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. J. Mach. Learn. Res. **9**(11) (2008)
23. Sundermeyer, M., Oparin, I., Gauvain, J.L., Freiberg, B., Schlüter, R., Ney, H.: Comparison of feedforward and recurrent neural network language models. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (2013)
24. Filus, K., Siavvas, M., Domańska, J., Gelenbe, E.: The random neural network as a bonding model for software vulnerability prediction. In: Calzarossa, M.C., Gelenbe, E., Grochla, K., Lent, R., Czachórski, T. (eds.) MASCOTS 2020. LNCS, vol. 12527, pp. 102–116. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68110-4_7
25. Sundermeyer, M., Schlüter, R., Ney, H.: LSTM neural networks for language modeling. In: Thirteenth Annual Conference of the ISCA (2012)

26. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv (2014)
27. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
28. Hochreiter, S.: The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertainty **6**, 107–116 (1998)

Correlation-Based Anomaly Detection for the CAN Bus

András Gazdag[1](✉), György Lupták[1], and Levente Buttyán[2]

[1] Laboratory of Cryptography and System Security (CrySyS Lab),
Department of Networked Systems and Services,
Budapest University of Technology and Economics, Budapest, Hungary
{agazdag,gyorgy.luptak}@crysys.hu

[2] Ukatemi Technologies, Budapest, Hungary
buttyan@ukatemi.com

Abstract. Previous attacks have shown that in-vehicle networks have vulnerabilities and a successful attack could lead to significant financial loss and danger to life. In this paper, we propose a Pearson correlation based anomaly detection algorithm to detect CAN message modification attacks. The algorithm does not need a priori information about the communication: it identifies signals based on statistical properties, finds the important correlation coefficients for the correlating signals, and detects attacks as deviations from a previously learned normal state.

Keywords: Controller area network · Anomaly detection · Correlation

1 Introduction

Modern vehicles are equipped with a large number of embedded controllers (ECUs) and other embedded computing devices, which are interconnected with networks internal to the vehicle (e.g., CAN buses, Ethernet), and some of these devices also have interfaces to external networks (e.g., Bluetooth, WiFi, 4G). The ECUs are responsible for various functions of the vehicle, some of which are safety critical. This setup makes vehicles subject to cyberattacks, whereby malicious actors may try to interfere with the behavior of the vehicle by accessing its internal components via its aforementioned external interfaces. The feasibility of such attacks on road vehicles have been demonstrated by researchers as a proof-of-concept [10], and there have been some real cases as well [16]. Unfortunately, attacks affecting safety critical functions may result in potentially fatal

The presented work was carried out within the MASPOV Project (KTI_KVIG_4-1_2021), which has been implemented with support provided by the Government of Hungary in the context of the Innovative Mobility Program of KTI. The research presented in this paper have also been supported by the NRDI Office, Ministry of Innovation and Technology, Hungary, within the framework of the Autonomous Systems National Laboratory Programme, and the NRDI Fund based on the charter of bolster issued by the NRDI Office.

E. Gelenbe et al. (Eds.): EuroCybersec 2021, CCIS 1596, pp. 38–50, 2022.
https://doi.org/10.1007/978-3-031-09357-9_4

accidents. Hence, it is clear that vehicles must be protected against cyberattacks, and the first step of this is to make them capable of detecting when they are attacked.

An overwhelming majority of the demonstrated attacks rely on injecting fake messages into the CAN bus. Some ECUs can be easily misled by the fake information in those injected illegitimate messages if they overweight the legitimate ones carrying the correct information. These injection attacks, however, are not so difficult to detect: one can observe the timing statistics of different types of messages and detect deviations from expected values by simple heuristic rules. For instance, [15] introduced the idea of analyzing the rate of messages for in-vehicle attack detection. In the normal state of vehicle operation, most message IDs appearing on the CAN bus have a regular frequency. When an attacker injects messages to achieve some malicious goal, the frequency of some message IDs will unexpectedly increase, as the legitimate ECUs will still send messages periodically with those message IDs besides the attacker's injected messages. Moreover, the frequency may be increased by a factor ranging from 2 to 100, as pointed out in [15]. Hence changes in frequencies can be detected quite easily by simple comparison to some fixed thresholds within a certain size of observation window. Equivalently, increased frequency of a message ID can be translated to decreased inter-arrival times for that message ID, and hence, changes in the statistics of inter-arrival times of certain message types can also serve as the basis for attack detection.

While the majority of attacks on the CAN bus indeed relies on message injection, this is not the only technique to achieve malicious goals. The predictability of message ID frequencies alone is not sufficient for detecting attacks in case of irregular or unpredictable CAN messages and in case of attacks that do not inject new messages on the CAN bus. In this paper, we address the latter problem: we propose a method to detect message modification attacks. Message modification attacks are more difficult to carry out, but they are also much more difficult to detect, therefore, attackers may consider them in the future, especially, given that message injection attacks will likely be detected by future vehicles. Achieving a message modification attack is difficult because the built in safety features of the CAN bus prevent a deliberate modification of a message on the fly. Hence, three options remain: (1) either the attacker compromises the relevant ECU to modify the message before it is even transmitted; (2) or a CAN gateway between two segments is compromised to modify a message during the hand-off between segments; (3) or the CAN bus is divided into two segments with a new malicious gateway inserted, allowing for message modifications on the gateways between the segments. The first two can be performed purely with software exploits, while the last requires physical modification of the network. Despite the increased complexity, the first and the last approaches to message modification attacks have already been demonstrated in [16] and in [3], respectively.

Our anomaly detection solution utilizes the fact that vehicle signals are correlated. The speed, the engine revolution, the current fuel consumption and many other values change together, representing the physical changes of the vehicle state. The solution proposed in this paper can represent these high level relationships between the vehicle signals with the correlation values that correspond to the normal state of the vehicle. During an attack, if a vehicle signal is overwritten by the attacker in a CAN message, some of the measured correlation values may deviate from those of the normal state, and this can be detected as an anomaly. An advantage of this approach, compared to previously proposed algorithms where only a single signal value is used in the anomaly detection, is that if the attacker does not modify all the correlating signals precisely at the same time, the attack can likely be detected.

The rest of the paper is structured as follows. Section 2 presents the related work. Section 3 introduces our proposed anomaly detection method and Sect. 1 summarizes its testing and validation. Finally, Sect. 5 concludes the paper.

2 Related Work

Academic papers proposing solutions for securing in-vehicle networks can be divided into three groups: (1) a relatively large body of work (e.g., [5,18,25]) is concerned with adding extensions to the CAN protocol, and by doing so, fixing its inherent security weaknesses; (2) a few papers (e.g., [14]) discuss intrusion prevention either by introducing firewalls or other techniques; and (3) another set of papers (which we discuss in more details below) deal with intrusion detection on the CAN bus using various approaches. Given the considerable amount of work done in the field, a few surveys have also been published: [9,19] have the broad scope of in-vehicle security as a whole, and [11,26,27] are more focused on in-vehicle intrusion detection. As our work falls in the domain of anomaly-based intrusion detection, we focus on that domain in the rest of this section.

Anomaly-based intrusion detection can take two approaches: *specification-based* and *model-based* anomaly detection. In case of the former, the normal behavior of the monitored system is explicitly specified. For instance, in the automotive case, the car manufacturer can have specifications for the normal frequency of different periodic messages at which they appear on the CAN bus. Deviations from the specification can easily be identified as signs of attack. In the case of model-based anomaly detection, on the other hand, no explicit specification of normal behavior is given, but instead, a model of the normal behavior is somehow obtained (e.g., learned by observing the system in the non-attacked state), and deviations from this model are detected as attacks. Different academic proposals differ in what modelling technique they use and what features of the system are modelled.

As for the modelling technique, many papers propose to use some statistical model (e.g., mean, variance, or entropy) of some parameter, with simple heuristic rules (e.g., comparison to a threshold) [4,15,17,20,21] or more sophisticated statistical hypothesis testing methods [12,24] to detect deviations from the model. Other papers (e.g., [7,13,22,23]) use some machine learning model (e.g., classifier or neural network), with the corresponding model specific method to decide if some input deviates from the learned model. Regarding the features that are modelled, many papers consider properties of the network traffic itself, such as packet timing features (e.g., frequency of certain types of packets) [17,20,21,24] and features related to the content of the packets (e.g., the time series of packet IDs or certain signal values) [8,12,13], whereas some papers use physical characteristics as features, such as voltage level and clock drift of physical signals on the CAN bus [2,6].

In this paper, we propose a model-based anomaly detection method for the CAN bus that uses correlations across different types of messages as features. To the best of our knowledge the only other paper using correlation-based anomaly detection is [1] therefore, we make a more in-depth comparison here. The method proposed in [1] computes the correlation between two specific signals, velocity and RPM, and detects attacks where extra messages containing incorrect values of these signals are injected into the bus. In contrast to this, our method computes the correlation between all pairs of signals, and identifies those pairs that have high correlation without identifying the actual signals. In addition, we detect message modification attacks, which are more difficult to detect than injection attacks. We simulate seven different types of modification attacks and evaluate the performance of our method for each of them. Otherwise, both [1] and our work use the Pearson correlation function, while in [1], the authors applied other analysis techniques (such as K-Means and Hidden Markov Models) as well.

3 Anomaly Detection Algorithm

3.1 Attacker Model

Our anomaly detection algorithm focuses only on the detection of message modification attacks, where the original repetition times of the messages are unchanged. As a result only the content of the messages can be used for anomaly detection.

3.2 Overview

We propose an anomaly detection algorithm that uses the correlation between signals encoded in CAN messages. Under normal conditions, the correlation between different signal pairs stays within a (signal pair specific) interval. In case of an attack where the attacker modifies only one member of a correlating signal pair, the resulting correlation may no longer stay within the interval, and this can be detected as an anomaly.

In the training phase, the correlation values between signals has to be determined. We measured multiple times the pairwise Pearson correlation between every signal pair in a one minute long time window and in a three minutes long time window. Next, based on these measurements, we decided whether the values are produced by an actual correlation. We achieved this by fitting different continuous probability distribution functions onto the measured correlation values. When we found a proper fit, we added the signal pair to our model. For every signal pair, we also calculated four thresholds to identify the boundaries of normal behaviour: (1) two thresholds define a narrow normal interval, such that measurement outside of this interval are considered potential anomalies for further analysis; (2) and another two thresholds define a wider interval, such that measurements outside of this interval are considered anomalies immediately.

In the detection phase, correlation values are determined in both a one minute long and a three minutes long window. Then the measured values are compared to the previously defined threshold for anomaly detection.

3.3 Data Preprocessing

In the training and testing phases we used a 31 min long CAN traffic log captured from a middle class vehicle. The traffic contains both periodic and non-periodic messages. This means that some messages arrive regularly with a fix repetition times, while others are only transmitted upon specific events. Before the training, we filtered out the non-periodic messages and those periodic messages that appear less than once per minute. After the filtering step, 92% of the original data remained.

The next step was the signal extraction from the traffic log. For this we used an algorithm from the Automated CAN Payload Reverse Engineering[1] project. This algorithm separates the bits of the CAN data field into signals based on bit flip frequencies, called Transition Aggregation N-Grams. The method builds on the property that the MSB bits of a signal change less frequently than the LSB bits. If there is a significant change in the bit flip frequencies of two neighboring bits that shows the boundary between two signals. The same statistical information can also be used to determine the signal encoding.

We calculated correlation values pairwise for the identified signals in different time windows. We tested window sizes from 1 to 8 min. For every interval, we re-sampled the signals to have two signal values per second within the chosen window. This rarefying speeds up the correlation calculation. Our measurements showed that the best results can be achieved by choosing 1 min and 3 min as final time windows. This allows us to detect significant anomalies fast and smaller anomalies in a reasonable time.

3.4 Model Training

Five matrices are calculated for both time windows for the purpose of anomaly detection. A matrix C contains the correlation values and there are four additional threshold matrices that store two upper ($C_{th+,1}$ $C_{th+,2}$) and two lower thresholds ($C_{th-,1}$ $C_{th-,2}$) for the detection.

Each cell $c_{i,j}$ in matrix C contains the Pearson correlation value calculated between signals i and j in the given time window. The correlation value is stored the following way:

[1] https://github.com/brent-stone/CAN_Reverse_Engineering.

- if the calculated coefficient indicates constant values then a NaN value is stored.
- otherwise if the two-tailed p-value of the Pearson correlation coefficient is less than or equal to 0.05, then the calculated value is stored.
- otherwise a 0 is stored.

During the training phase, we randomly select a starting point in the CAN log and calculate the correlation values for all signal pairs for both time intervals. Selecting the starting point randomly allows us to use the original trace multiple times generating a correlation matrix with small differences for every starting point. With this method, we created 300 training matrices for the threshold calculations.

These training matrices gave us a good representation of the typical correlation value for all pairs of signals. In order to find thresholds characterizing the normal behaviour, we fitted different continuous probability distribution functions onto the $c_{i,j}$ values of every training matrix. For every distribution, we performed a Kolmogorov-Smirnov (K-S) test to find the distribution that fits best the correlation values. The K-S test gave us two results: a D statistics and a p-value. For the former, we calculated[2] a critical value at significance level $\alpha = 1\%$ using (1) (where n is the number of samples in the dataset):

$$d_{1\%} = \frac{1.6276}{\sqrt{n}} \tag{1}$$

Those distributions, where the resulting D statistic was less than or equal to the critical value and the resulting fitted probability distribution's standard deviation was greater than 0 and less than 0.2, we accepted the distribution as a potential candidate. Then, for all these candidates, we calculated the probability distribution's percent-point (or quantile) function value for the $10^{-3}, 1 - 10^{-3}, 10^{-6}, 1 - 10^{-6}$ probabilities. These gave us candidates for the minimum ($min_{i,j}$), maximum ($max_{i,j}$), significant minimum ($sigmin_{i,j}$) and significant maximum ($sigmax_{i,j}$) thresholds.

To choose the best option from the candidates, a scoring technique was used, which is based on the length of the normalized significant min-max interval ($sigmax_{i,j} - sigmin_{i,j}$) and the normalized min-max intervals ($max_{i,j} - min_{i,j}$). The candidate with the minimum final score was selected as the final probability distribution. We used the following function (Fig. 1 and Eq. 2) with a minimum value of 0.6 for the significant min-max, and a minimum value of 0.5 for the min-max intervals:

$$score(x) = \begin{cases} (x - min)^2, \text{if } x \leqq min \\ \frac{x-min}{min*x}, \text{if } x > min \end{cases} \tag{2}$$

Then, for the final candidate score we used formula (3):

$$final\_score = 0.65 * minmax\_score + 0.35 * significant\_minmax\_score \tag{3}$$

[2] https://www.real-statistics.com/statistics-tables/kolmogorov-smirnov-table/.

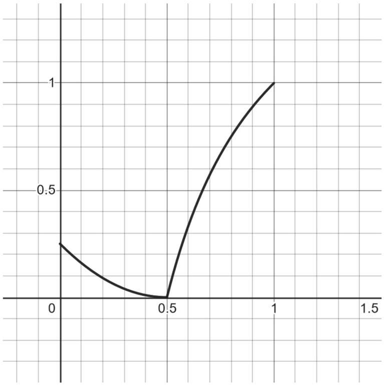


Fig. 1. Visualization of the function used to score the normal min-max threshold intervals.

The used scoring system assigns the smallest score to the candidates with a min-max interval length closest to 0.5 and a significant min-max interval length closest to 0.6. This approach prefers the candidates where the intervals are relatively small but not too tight. Our measurements showed that these typically used statistical constants in the calculations with this weighting gives the best trade-off between false positive and false negative results. We used a larger weight for the min-max score to reflect that it is more important to detect and attack the speed of the detection.

These final chosen threshold values are stored in the threshold matrices, given that there was at least one candidate distribution, in the following way: the minimum values are stored in matrix $C_{th-,1}$; the maximum values are stored in matrix $C_{th+,1}$; the significant minimum values are stored in matrix $C_{th-,2}$, and finally, the significant maximum values are stored in matrix $C_{th+,2}$. If there was no candidate probability distribution found, the signal pair was excluded from the study.

In Fig. 2, we present an example of the measured correlation values of a signal pair and the fitted probability distribution function (‘loggamma’). The vertical blue and red lines show the determined minimum and maximum, and significant minimum and maximum values, respectively.

3.5 Detection

In the detection phase, the current correlation values are calculated for the last 1 and 3 min of the network traffic, and then, the results are compared to the threshold matrices in the following way:

- if a correlation value is outside the significant min-max interval, it is immediately detected as an attack;

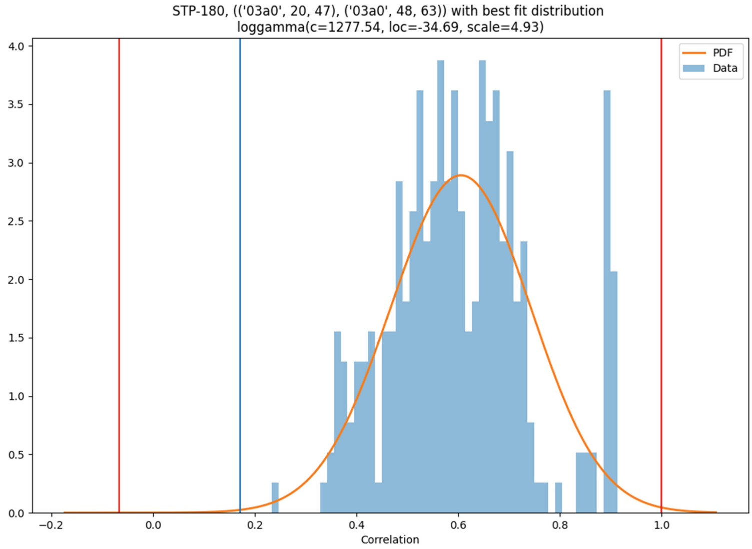


Fig. 2. One example of a signal pair with a fitted 'loggamma' probability distribution

- if the value is only outside the normal min-max interval, but not outside the significant one, we consider it a potential attack only, which will be a real attack if our model with the other time interval also gets a similar result.

4 Evaluation of the Algorithm

4.1 Testing

In the testing phase, we took 270 1-min long samples of the original trace, which we also used for training. We performed the detection step of the algorithm on these samples with a previously trained model and found the following result: the model signaled an attack in 14 out of 270 cases, resulting in a false positive rate of 5.2%.

4.2 Validation

Infected Dataset. The validation of the proposed algorithm was performed on an infected dataset. We simulated different message modification attacks (no new message is added to the log) with a previously developed attack simulator[3] that can take a clean CAN log and modify a selected subset (specified by ID and time interval) of its messages according to 7 different attack scenarios:

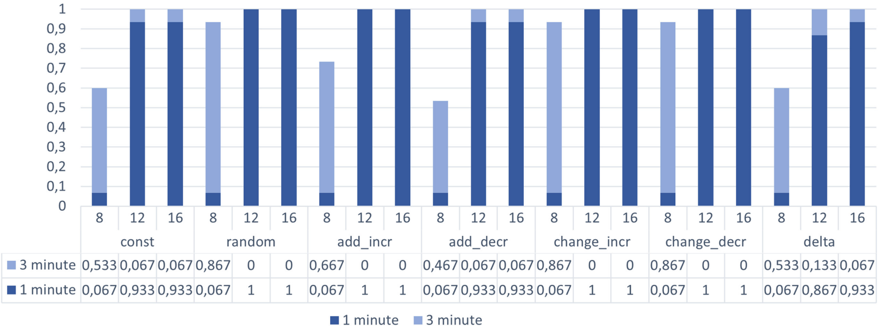
1. **const:** the original data value is replaced by a given attack data.

[3] https://github.com/CrySyS/can-log-infector.

2. **random:** the original data value is replaced by a new random value.
3. **delta:** a given attack data is added to the original data value.
4. **add_incr:** an increasing value is added to the original data value.
5. **add_decr:** an increasing value is subtracted from the original value.
6. **change_incr:** the original data value is replaced by an increasing value.
7. **change_decr:** the original data value is replaced by a decreasing value.

Measurements. In order to evaluate the performance of the algorithm in more details, we divided the signals into three different groups and validated the algorithm in each group separately. The first group contain signals that strongly correlate with multiple other signals. Typically, the most important signals of a vehicle belong to this group. The second group contains signals that have a strong correlation with one other signal, and the third group contains signals with only weak correlation values. We considered a correlation strong between two signals if the mean of the absolute correlation value was above or equal to 0.9 for all the 300 training data samples.

We chose from each group 4 or 5 signal pairs for the validation. For these signals, we simulated all previously mentioned 7 attacks on 15 randomly chosen segments of the original trace. Each attack was performed multiple times. First, only 8 bits was modified according to the attack description in one of the target signals, than the number of affected bits in the upcoming test was increased by 4 until the signal length was reached. All of the attacks were theoretical, but based on previous real life attack descriptions. We did not check whether a specific attack would actually have any impact in real life.

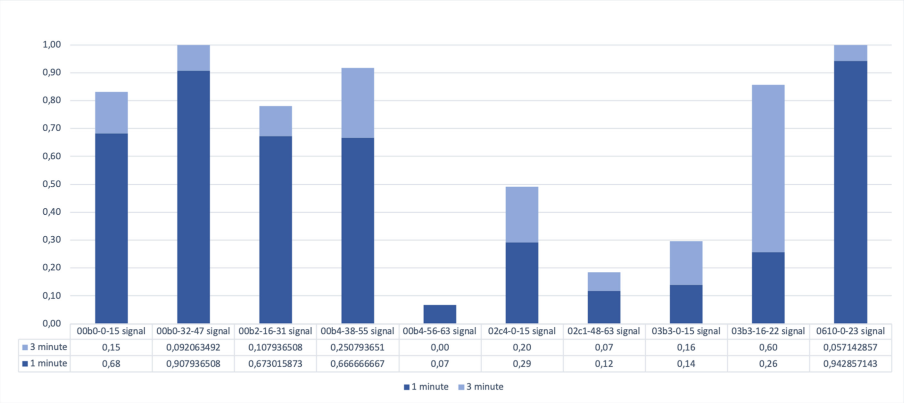


	8	12	16	8	12	16	8	12	16	8	12	16	8	12	16	8	12	16	8	12	16
	const			random			add_incr			add_decr			change_incr			change_decr			delta		
▨ 3 minute	0,533	0,067	0,067	0,867	0	0	0,667	0	0	0,467	0,067	0,067	0,867	0	0	0,867	0	0	0,533	0,133	0,067
■ 1 minute	0,067	0,933	0,933	0,067	1	1	0,067	1	1	0,067	0,933	0,933	0,067	1	1	0,067	1	1	0,067	0,867	0,933

Fig. 3. Testing results for 16 bit long signal with strong correlations.

Figure 3 shows detailed results for a signal with strong correlations. The 16 bit long signal was attacked with all attack types. For each type, 3 attacks were performed where the affected number of bits increase from 8 to 16. The two colors of the columns indicate which time window was successful for the attack detection. The detection rate varies between 55% and 100% with an above 90% result for attacks modifying more than 12 bits.

The results found in the others groups, as expected, are less accurate. The average detection accuracy of attacks of signals with one strong correlation is 58% while this falls to 20% for the third group where the signals only have weak correlations.



	00b0-0-15 signal	00b0-32-47 signal	00b2-16-31 signal	00b4-38-55 signal	00b4-56-63 signal	02c4-0-15 signal	02c1-48-63 signal	03b3-0-15 signal	03b3-16-22 signal	0610-0-23 signal
■ 3 minute	0,15	0,092063492	0,107936508	0,250793651	0,00	0,20	0,07	0,16	0,60	0,057142857
■ 1 minute	0,68	0,907936508	0,673015873	0,666666667	0,07	0,29	0,12	0,14	0,26	0,942857143

Fig. 4. Detection accuracy of high priority signals.

Figure 4 shows a summary of the results for messages with the highest priority (based on the CAN ID field). It can be seen, that most messages with the highest priority contain signals with high correlation, making them ideal candidates for a correlation based anomaly detection.

5 Conclusion and Future Work

In this paper, we proposed a novel correlation based anomaly detection method for the CAN bus with a focus on message modification attacks. We showed, that our solution efficiently detects most of the attacks, making it a promising candidate for real life anomaly detection. Furthermore, a significant advantage of this correlation based detection is that it can detect even the most sophisticated attacks, assuming that the attacker does not modify every related signals consistently.

In our future work, we plan to investigate if the proposed threshold based detection mechanism can be replaced with other potential solutions that increase accuracy. A potential option for this is a machine learning based classification. Moreover, the efficiency of the correlation calculation could also be increased with better data preprocessing, in order to further improve the applicability of our solution in real life scenarios.

References

1. Ben Othmane, L., Dhulipala, L., Abdelkhalek, M., Multari, N., Govindarasu, M.: On the performance of detecting injection of fabricated messages into the can bus. IEEE Trans. Dependable Secure Comput., 1 (2020). https://doi.org/10.1109/TDSC.2020.2990192
2. Choi, W., Joo, K., Jo, H.J., Park, M.C., Lee, D.H.: VoltageIDS: low-level communication characteristics for automotive intrusion detection system. IEEE Trans. Inf. Forensics Secur. **13**(8), 2114–2129 (2018). https://doi.org/10.1109/TIFS.2018.2812149
3. Gazdag, A., Ferenczi, C., Buttyán, L.: Development of a man-in-the-middle attack device for the can bus. In: Proceedings of the 1st Conference on Information Technology and Data Science Debrecen, Hungary, 6–8 November 2020, pp. 115–130 (2020)
4. Gmiden, M., Mohamed, H., Trabelsi, H.: An intrusion detection method for securing in-vehicle CAN bus. In: Proceedings of the 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA) (2016). https://doi.org/10.1109/STA.2016.7952095
5. Groza, B., Murvay, S., van Herrewege, A., Verbauwhede, I.: LiBrA-CAN: a lightweight broadcast authentication protocol for controller area networks. In: Pieprzyk, J., Sadeghi, A.-R., Manulis, M. (eds.) CANS 2012. LNCS, vol. 7712, pp. 185–200. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-35404-5_15
6. Ji, H., Wang, Y., Qin, H., Wu, X., Yu, G.: Investigating the effects of attack detection for in-vehicle networks based on clock drift of ECUs. IEEE Access **6**, 49375–49384 (2018). https://doi.org/10.1109/ACCESS.2018.2841884
7. Kang, M.J., Kang, J.W.: Intrusion detection system using deep neural network for in-vehicle network security. PLoS One **11**(6), e0155781 (2016)
8. Kang, M.J., Kang, J.W.: A novel intrusion detection method using deep neural network for in-vehicle network security. In: Proceedings of the 83rd IEEE Vehicular Technology Conference (VTC Spring), pp. 1–5 (2016). https://doi.org/10.1109/VTCSpring.2016.7504089
9. Kim, K., Kim, J.S., Jeong, S., Park, J.H., Kim, H.K.: Cybersecurity for autonomous vehicles: review of attacks and defense. Comput. Secur. **103** (2021). https://doi.org/10.1016/j.cose.2020.102150
10. Koscher, K., et al.: Experimental security analysis of a modern automobile. In: 2010 IEEE Symposium on Security and Privacy, pp. 447–462 (2010). https://doi.org/10.1109/SP.2010.34
11. Lokman, S.-F., Othman, A.T., Abu-Bakar, M.-H.: Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review. EURASIP J. Wirel. Commun. Netw. **2019**(1), 1–17 (2019). https://doi.org/10.1186/s13638-019-1484-3
12. Marchetti, M., Stabili, D.: Anomaly detection of can bus messages through analysis of id sequences. In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV), pp. 1577–1583 (2017). https://doi.org/10.1109/IVS.2017.7995934
13. Markovitz, M., Wool, A.: Field classification, modeling and anomaly detection in unknown CAN bus networks. In: Proceedings of the Embedded Security in CARs (ESCAR) Conference (2015)

14. Matsumoto, T., Hata, M., Tanabe, M., Yoshioka, K., Oishi, K.: A method of preventing unauthorized data transmission in controller area network. In: Proceedings of the 75th IEEE Vehicular Technology Conference (VTC Spring), pp. 1–5 (2012). https://doi.org/10.1109/VETECS.2012.6240294
15. Miller, C., Valasek, C.: A survey of remote automotive attack surfaces. Technical report, IOActive (2014). https://ioactive.com/pdfs/IOActive_Remote_Attack_Surfaces.pdf
16. Miller, C., Valasek, C.: Adventures in automotive networks and control units. Technical report, IOActive (2013)
17. Moore, M.R., Bridges, R.A., Combs, F.L., Starr, M.S., Prowell, S.J.: Modeling inter-signal arrival times for accurate detection of CAN bus signal injection attacks. In: Proceedings of the 12th Annual Conference on Cyber and Information Security Research (2017)
18. Nürnberger, S., Rossow, C.: vatiCAN - vetted, authenticated CAN bus. In: Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems (CHES), pp. 106–124 (2016)
19. Sharma, C., Moylan, S., Amariucai, G.T., Vasserman, E.Y.: An extended survey on vehicle security. Computing Research Repository (CoRR) abs/1910.04150 (2019). http://arxiv.org/abs/1910.04150
20. Song, H.M., Kim, H.R., Kim, H.K.: Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In: Proceedings of the International Conference on Information Networking (ICOIN) (2016). https://doi.org/10.1109/ICOIN.2016.7427089
21. Taylor, A., Japkowicz, N., Leblanc, S.: Frequency-based anomaly detection for the automotive CAN bus. In: Proceedings of the World Congress on Industrial Control System Security (WCICSS), December 2015
22. Taylor, A., Leblanc, S., Japkowicz, N.: Anomaly detection in automobile control network data with long short-term memory networks. In: Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 130–139 (2016). https://doi.org/10.1109/DSAA.2016.20
23. Theissler, A.: Anomaly detection in recordings from in-vehicle networks. In: Proceedings of the International Workshop on Big Data Applications and Principles (2014)
24. Tomlinson, A., Bryans, J., Shaikh, S.A., Kalutarage, H.K.: Detection of automotive CAN cyber-attacks by identifying packet timing anomalies in time windows. In: Proceedings of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), pp. 231–238 (2018). https://doi.org/10.1109/DSN-W.2018.00069
25. Van Herrewege, A., Singelée, D., Verbauwhede, I.: CANAuth - a simple, backward compatible broadcast authentication protocol for CAN bus. In: Proceedings of the ESCAR Conference (2011)
26. Wu, W., et al.: A survey of intrusion detection for in-vehicle networks. IEEE Trans. Intell. Transp. Syst. **21**(3) (2020). https://doi.org/10.1109/TITS.2019.2908074
27. Young, C., Zambreno, J., Olufowobi, H., Bloom, G.: Survey of automotive controller area network intrusion-detection systems. IEEE Des. Test (2019). https://doi.org/10.1109/MDAT.2019.2899062

Botnet Attack Detection with Incremental Online Learning

Mert Nakip[1](✉) and Erol Gelenbe[1,2,3,4]

[1] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,
44-100 Gliwice, Poland
{mnakip,seg}@iitis.pl
[2] Yaşar University, Bornova/İzmir, Turkey
[3] Lab. I3S Université, Côte d'Azur, 06200 Nice, France
[4] Lab. A. De Moivre CNRS, London, UK

Abstract. In recent years, IoT devices have often been the target of Mirai Botnet attacks. This paper develops an intrusion detection method based on Auto-Associated Dense Random Neural Network with incremental online learning, targeting the detection of Mirai Botnet attacks. The proposed method is trained only on benign IoT traffic while the IoT network is online; therefore, it does not require any data collection on benign or attack traffic. Experimental results on a publicly available dataset have shown that the performance of this method is considerably high and very close to that of the same neural network model with offline training. In addition, both the training and execution times of the proposed method are highly acceptable for real-time attack detection.

Keywords: Internet of Things (IoT) · Botnet attacks · Mirai · Incremental learning · Auto associative neural networks · Dense random neural networks

1 Introduction

Since IoT devices in the Massive IoT segment are low-cost devices, they can often perform a single task at a time and their computational power is not sufficient to execute complex attack detection algorithms. Therefore, Massive IoT is vulnerable to network attacks. According to a study by HP [1], 70% of IoT devices are vulnerable to attacks, while one of the most common attacks is the Denial of Service (DoS) attack which comprises 20% of all attacks against the IoT [6].

Network attacks can include worms based on propagating software [37,38], DoS attacks where an attacker or an infected device aims to prevent the normal functioning of a device (or a system) by forwarding superfluous requests [10,11], and Botnets which are the subject of this work. Traffic which may cause attacks can be detected as a form of anomaly [22,26] which is concealed as part of normal innocuous traffic.

E. Gelenbe et al. (Eds.): EuroCybersec 2021, CCIS 1596, pp. 51–60, 2022.
https://doi.org/10.1007/978-3-031-09357-9_5

When a network attack occurs following the same techniques as DoS attacks, but affecting more devices it is called a Distributed DoS (DDoS) attack [14]. One of the most popular kinds of DDoS attacks is the Botnet attack which mainly targets IoT devices. In a Botnet attack, a victim device turns into a bot via malware and generates traffic that floods other servers and devices with meaningless requests that lead to threats [21].

Detecting Botnet attacks is an important task considering the high threat level for a massive number of devices. To this end, a recent trend of research has focused on developing Machine Learning (ML) based techniques. Most of earlier work [5,13,24,28–30,34–36,39,40,42] in this trend develops techniques for classification by supervised learning; however, these techniques require large numbers of samples for both normal traffic and malicious traffic; collecting data for realistic malicious traffic is no easy task. Only a few works evaluated the lack of attack data during the training of ML models (via auto-associative learning) for Botnet attacks [33,43] and for DoS attacks [15].

In 2016, a massive DDoS, Botnet, attack affected many web sites including Netflix, Reddit, Spotify, and Twitter through the Dyn service for domain name system (DNS) management [7,23] as well as numerous IP addresses creating access through the servers of some cyber-security companies [41]. It is known that the botnets in this DDoS attack were infected by the Mirai malware, in which the infected devices generate traffic that overwhelms servers and other devices with nonsense requests, sometimes leading to threats [21]. Reference [4] has analyzed the characteristics of this class of attacks, while a recent work [27] has analyzed the characteristics of IoT traffic generated by Botnet. In addition, Reference [3] used blockchains to protect IoT networks against Mirai Botnet attacks.

1.1 Attack Detection with the Random Neural Network (RNN)

The RNN [17] with gradient descent learning [18] has been used to detect Denial of Service attacks in early work [34] and was recently used also to detect SYN attacks [15].

The Dense RNN was introduced in [16,20] to address various pattern recognition problems, including character and object recognition. It has been previously used with auto-associative offline training to detect SYN attacks [8], and was used more recently also to detect Mirai Botnet attacks [33].

In this paper, we use a Dense Random Neural Network (Dense RNN) [16,20] based Mirai Botnet attack detection method, but extend it specifically for incremental online learning. Similar to [8], this method learns the statistics of the IoT traffic under normal circumstances while the network is online (via auto-associative and incremental online learning); that is, it does not require the offline collection of any IoT traffic data (either benign or attack) for the learning procedure.

In the rest of this paper, Sect. 2 presents the methodology of the proposed method for Mirai Botnet attack detection while Sect. 4 presents the performance evaluation of this method on a publicly available dataset. Lastly, Sect. 6 summarizes the paper.

2 Auto-Associative Dense RNN Based Botnet Attack Detection with Online Incremental Training

We now present the methodology of our Botnet Attack Detector (AD) based on Dense Random Neural Networks (Dense RNN) which is trained entirely online with only benign IoT traffic. Figure 1 displays the architectural design of this detector, which consists of three main stages:

1. Extracting metrics from IoT traffic with the “Metric Extractor” module,
2. Detection of potential attack packets with “Auto-Associative Dense RNN” and “Attack Decision Maker” modules and
3. Incremental online training of AA-Dense RNN with “Incremental Semi-Supervised Learning Algorithm”. In the rest of this section, we shall detail the methodologies of these stages.

3 Extracting Metrics from IoT Traffic

Considering that the Mirai botnet attacks aim to spread through the devices in the IoT network, a recent work [33] has proposed three metrics calculated using only the transmission times and lengths of packets. Since the correlation analysis presented in [33] has shown that these three metrics successfully captures the traces of Mirai botnet attack packets, this paper also uses these metrics, which are defined as follows:

- **Metric 1**: The total size of the last N transmitted packets,
- **Metric 2**: The average inter-transmission times of the last N packets,
- **Metric 3**: Total number of packets that are transmitted in last T seconds.

Furthermore, it has also been shown that an attack detector achieves its best performance using these metrics with importance coefficients. However, in order to design an attack detector with purely online training on only normal unlabeled traffic, we will treat these metrics equally, i.e. take each of their importance coefficients as $1/3$.

The Dense-RNN model, which allows direct connectivity between neuron cells (addition to the usual axon-dendrite interactions), has been proposed in [16,20]. It is a specific form of the Random Neural Network (RNN) [12,17] that uses clusters of RNN cells for deep learning.

Earlier research have shown the success of the conventional RNN model [19] in IoT systems for applications on the video quality evaluation [32], network design [9], and home climate control [25]. In the Dense RNN model, firing at any cell may trigger a direct firing at a neighboring cell as well as excite or inhibit any other cell in the neural network through corresponding weights. In addition, probability p that any other cell in the network fires when a given cell fires, represents the direct interaction between neuron cells.

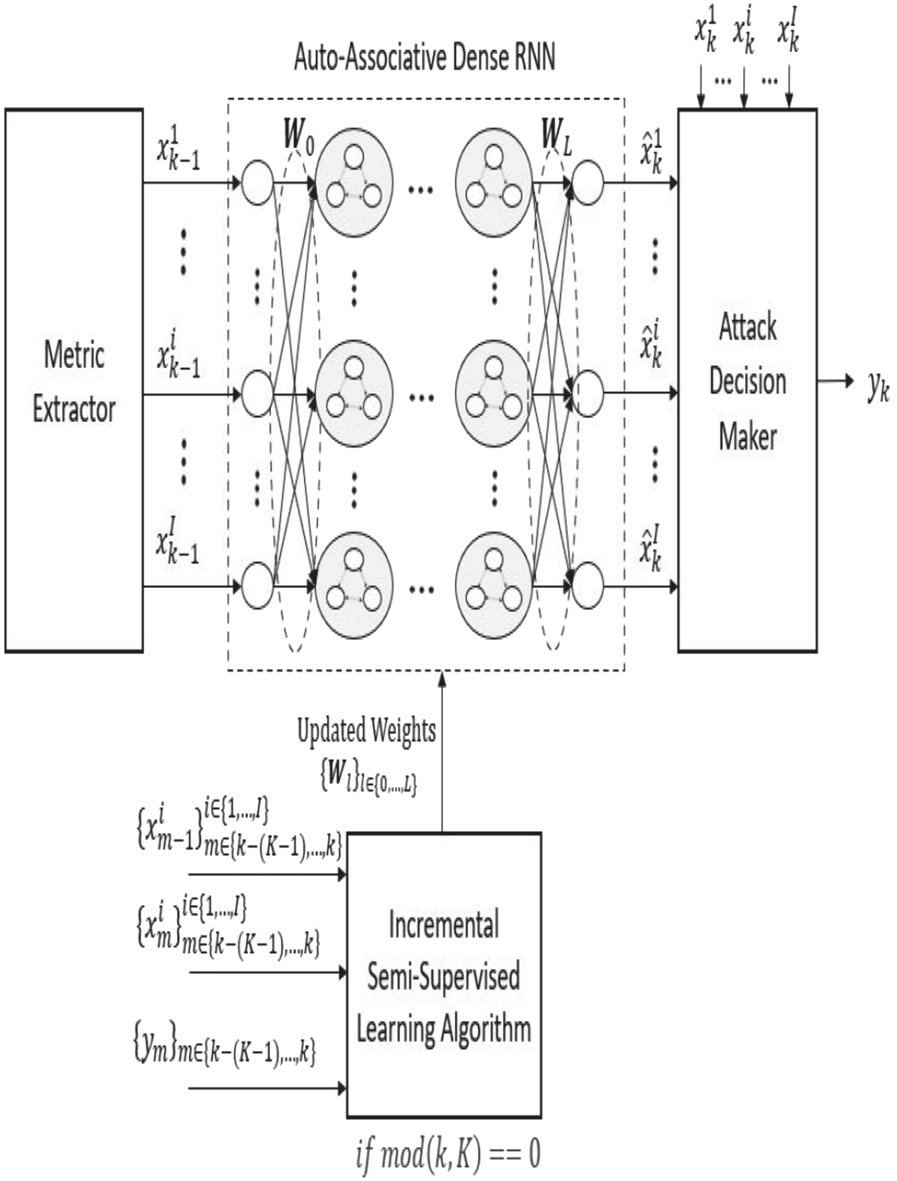

Auto-Associative Dense RNN
Metric Extractor
x_{k-1}^1
x_{k-1}^i
x_{k-1}^I
W_0
W_L
$\hat{x}_k^1$
$\hat{x}_k^i$
$\hat{x}_k^I$
x_k^1 x_k^i x_k^I
Attack Decision Maker
y_k
Updated Weights $\{W_l\}_{l\in\{0,\dots,L\}}$
$\{x_{m-1}^i\}_{m\in\{k-(K-1),\dots,k\}}^{i\in\{1,\dots,I\}}$
$\{x_m^i\}_{m\in\{k-(K-1),\dots,k\}}^{i\in\{1,\dots,I\}}$
$\{y_m\}_{m\in\{k-(K-1),\dots,k\}}$
Incremental Semi-Supervised Learning Algorithm
$if\ mod(k,K) == 0$

Fig. 1. The architectural design of AA-Dense RNN based Botnet attack detector with incremental online training

4 Experimental Results

In order to evaluate the performance of our AA-Dense RNN based Botnet attack detector with incremental online learning, we use publicly-available Kitsune dataset [2,31] which contains 764, 137 normal and malicious packets for Mirai Botnet attack. During the performance evaluation, we compare the performance of AA-Dense RNN under online training with that under offline training.

For AA-Dense RNN, we first set the number of neurons in each layer l as $n_l = I = 3$, and $p = 0.05$, $r = 0.001$ and $\lambda^+ = \lambda^- = 0.1$. We also set $N = 500$ packets and $T = 100$ secs for the extraction of metrics, and we set $\Theta = 0.02$ for Attack Decision Maker module.

First, we evaluate the performance of the proposed AD method for varying number of training packets K between 100 and 1000. In this way, we shall also select the best value of K and set it for the rest of this section.

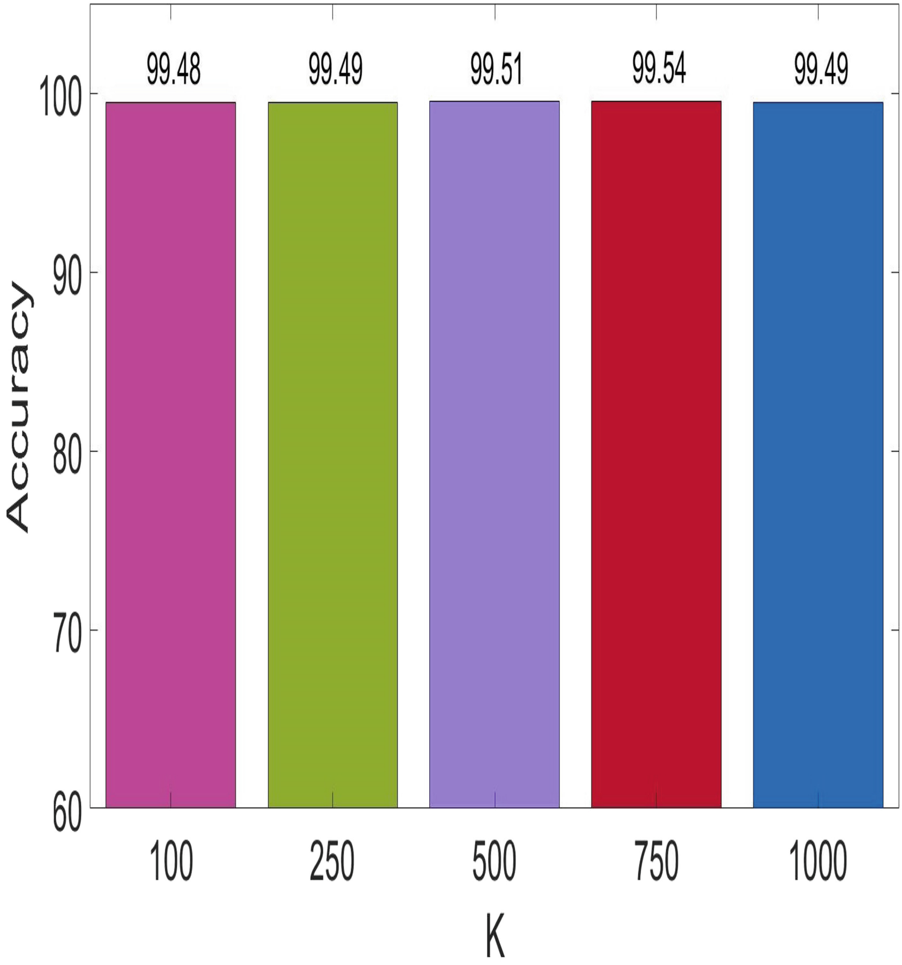


Fig. 2. Average accuracy of AA-Dense RNN attack detector with incremental online learning for different values of $K \in \{100, 250, 500, 750, 1000\}$

Figure 2 presents the average classification accuracy (over all packets) for each value of $K \in \{100, 250, 500, 750, 1000\}$. The results in Fig. 2 show that AA-Dense RNN with incremental online learning achieves its best performance for $K = 750$ packets, where the average accuracy equals 99.54. In addition, one may see that AA-Dense RNN achieves acceptable accuracy for all K.

5 Computation Time

For the proposed method, Table 1 presents the execution time (i.e. time elapsed) for making a decision on a single packet as well as the initialization and incremental update stages of the training algorithm for $K = 750$. Note that we measured the computation times on a PC with 32 GB ram and AMD Ryzen 7 3.70 GHz processor.

The results in this table first show that the execution time of AA-Dense RNN is very low and acceptable for real-time attack detection. Also, we see that the initialization and incremental online learning of our method take 15 ms and 4.3 ms, respectively. As observed in the evaluated dataset, 4.3 ms is slightly less than the minimum measured time for transmission of 22 packets; that is, the parameters of AA-Dense RNN will be updated until the transmission of the $22nd$ packet after the incremental online learning phase has begun.

Table 1. Training and online run-times of the proposed attack detection method with incremental online learning

Training time (for $K = 750$)	Initialization	15 ms
	Incremental update	4.3 ms
Execution time		0.11 ms

6 Conclusions

Devices in the Massive IoT segment are vulnerable targets for Mirai Botnet attacks as they are often deployed quickly with low-security measures. Therefore, in this paper, we developed a Mirai Botnet attack detection method based on Auto-Associative Dense Random Neural Network (AA-Dense RNN) with an incremental online learning algorithm. One of the main advantages of this method is that it learns the statistics of the normal (benign) IoT traffic when the IoT network is online, so it does not require collecting any (benign or attack) traffic beforehand.

We have evaluated the performance of the proposed method on a publicly available dataset containing 764, 137 packet transmissions and compared the performance of the proposed online AA-Dense RNN based attack detection method with that of offline trained AA-Dense RNN.

Our experimental results show that the proposed method achieves 99.54% accuracy with 99.79% TPR and 98.19% TNR while both training time (initialization and update) and execution time are very small and highly acceptable for real-time lightweight Mirai Botnet attack detection.

Our future work will extend our design to detect the various type of attacks via a single detector with incremental online training on only benign IoT traffic.

Acknowledgments. This research has been supported by the European Commission H2020 Program through the IoTAC Research and Innovation Action, under Grant Agreement No. 952684.

References

1. Hp study reveals 70 percent of Internet of Things devices vulnerable to attack. https://www.hp.com/us-en/hp-news/press-release.html?id=1744676
2. Kitsune Network Attack Dataset, August 2020. https://www.kaggle.com/ymirsky/network-attack-dataset-kitsune
3. Ahmed, Z., Danish, S.M., Qureshi, H.K., Lestas, M.: Protecting IoTs from Mirai Botnet attacks using blockchains. In: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 1–6 (2019). https://doi.org/10.1109/CAMAD.2019.8858484
4. Antonakakis, M., et al.: Understanding the Mirai Botnet. In: Proceedings of the 26th USENIX Security Symposium (2017). https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis
5. Banerjee, M., Samantaray, S.: Network traffic analysis based iot botnet detection using honeynet data applying classification techniques. Int. J. Comput. Sci. Inf. Secur. (IJCSIS) **17**(8) (2019)
6. Benzarti, S., Triki, B., Korbaa, O.: A survey on attacks in Internet of Things based networks. In: 2017 International Conference on Engineering & MIS (ICEMIS), pp. 1–7. IEEE (2017)
7. Biggs, J.: Hackers release source code for a powerful DDoS app called Mirai. TechCrunch, October 2018. https://techcrunch.com/2016/10/10/hackers-release-source-code-for-a-powerful-ddos-app-called-mirai/
8. Brun, O., Yin, Y., Gelenbe, E., Kadioglu, Y.M., Augusto-Gonzalez, J., Ramos, M.: Deep learning with dense random neural networks for detecting attacks against IoT-connected home environments. In: Gelenbe, E., et al. (eds.) Euro-CYBERSEC 2018. CCIS, vol. 821, pp. 79–89. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-95189-8_8
9. Cancela, H., Robledo, F., Rubino, G.: A grasp algorithm with RNN based local search for designing a wan access network. Electron. Not. Discrete Math. **18**, 59–65 (2004). https://doi.org/10.1016/j.endm.2004.06.010. https://www.sciencedirect.com/science/article/pii/S1571065304010674
10. Carl, G., Kesidis, G., Brooks, R., Rai, S.: Denial-of-service attack-detection techniques. IEEE Internet Comput. **10**(1), 82–89 (2006). https://doi.org/10.1109/MIC.2006.5
11. CISA: Understanding Denial-of-Service attacks. https://us-cert.cisa.gov/ncas/tips/ST04-015

12. Cramer, C.E., Gelenbe, E.: Video quality and traffic QoS in learning-based subsampled and receiver-interpolated video sequences. IEEE J. Sel. Areas Commun. **18**(2), 150–167 (2000)
13. Doshi, R., Apthorpe, N., Feamster, N.: Machine learning DDoS detection for consumer internet of things devices. In: 2018 IEEE Security and Privacy Workshops (SPW), pp. 29–35. IEEE (2018)
14. Douligeris, C., Mitrokotsa, A.: DDoS attacks and defense mechanisms: classification and state-of-the-art. Comput. Netw. **44**(5), 643–666 (2004)
15. Evmorfos, S., Vlachodimitropoulos, G., Bakalos, N., Gelenbe, E.: Neural network architectures for the detection of SYN flood attacks in IoT systems. In: Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments, pp. 1–4 (2020)
16. Gelenbe, E., Yin, Y.: Deep learning with random neural networks. In: 2016 International Joint Conference on Neural Networks (IJCNN), pp. 1633–1638 (2016). https://doi.org/10.1109/IJCNN.2016.7727393
17. Gelenbe, E.: Random neural networks with negative and positive signals and product form solution. Neural Comput. **1**(4), 502–510 (1989)
18. Gelenbe, E.: Learning in the recurrent random neural network. Neural Comput. **5**(1), 154–164 (1993)
19. Gelenbe, E., Stafylopatis, A.: Global behavior of homogeneous random neural systems. Appl. Math. Model. **15**(10), 534–541 (1991)
20. Gelenbe, E., Yin, Y.: Deep learning with dense random neural networks. In: Gruca, A., Czachórski, T., Harezlak, K., Kozielski, S., Piotrowska, A. (eds.) ICMMI 2017. AISC, vol. 659, pp. 3–18. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-67792-7_1
21. Goodin, D.: 100,000-strong Botnet built on router 0-day could strike at any time. Ars Technica, December 2017. https://arstechnica.com/information-technology/2017/12/100000-strong-botnet-built-on-router-0-day-could-strike-at-any-time/
22. Grenet, I., Yin, Y., Comet, J.-P., Gelenbe, E.: Machine learning to predict toxicity of compounds. In: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I. (eds.) ICANN 2018. LNCS, vol. 11139, pp. 335–345. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01418-6_33
23. Hackett, R.: Why a hacker dumped code behind colossal website-trampling botnet, October 2016
24. Htwe, C.S., Thant, Y.M., Thwin, M.M.S.: Botnets attack detection using machine learning approach for IoT environment. J. Phys. Conf. Ser. **1646**, 012101 (2020)
25. Javed, A., Larijani, H., Ahmadinia, A., Gibson, D.: Smart random neural network controller for HVAC using cloud computing technology. IEEE Trans. Industr. Inf. **13**, 351–360 (2017)
26. Kim, H., Gelenbe, E.: Anomaly detection in gene expression via stochastic models of gene regulatory networks. In: BMC Genomics, vol. 10, pp. 1–10. BioMed Central (2009)
27. Kumar, A., Lim, T.J.: Early detection of Mirai-like IoT bots in large-scale networks through sub-sampled packet traffic analysis. In: Arai, K., Bhatia, R. (eds.) FICC 2019. LNNS, vol. 70, pp. 847–867. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-12385-7_58
28. Letteri, I., Del Rosso, M., Caianiello, P., Cassioli, D.: Performance of botnet detection by neural networks in software-defined networks. In: ITASEC (2018)
29. Liu, J., Liu, S., Zhang, S.: Detection of IoT botnet based on deep learning. In: 2019 Chinese Control Conference (CCC), pp. 8381–8385. IEEE (2019)

30. McDermott, C.D., Majdani, F., Petrovski, A.V.: Botnet detection in the Internet of Things using deep learning approaches. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2018)
31. Mirsky, Y., Doitshman, T., Elovici, Y., Shabtai, A.: Kitsune: an ensemble of autoencoders for online network intrusion detection. In: The Network and Distributed System Security Symposium (NDSS) 2018 (2018)
32. Mohamed, S., Rubino, G.: A study of real-time packet video quality using random neural networks. IEEE Trans. Circuits Syst. Video Technol. **12**(12), 1071–1083 (2002)
33. Nakip, M., Gelenbe, E.: MIRAI botnet attack detection with auto-associative dense random neural network. In: IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2021)
34. Oke, G., Loukas, G., Gelenbe, E.: Detecting denial of service attacks with Bayesian classifiers and the random neural network. In: 2007 IEEE International Fuzzy Systems Conference, pp. 1–6. IEEE (2007)
35. Parra, G.D.L.T., Rad, P., Choo, K.K.R., Beebe, N.: Detecting Internet of Things attacks using distributed deep learning. J. Netw. Comput. Appl. **163**, 102662 (2020)
36. Prokofiev, A.O., Smirnova, Y.S., Surov, V.A.: A method to detect internet of things botnets. In: 2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), pp. 105–108. IEEE (2018)
37. Sakellari, G., Gelenbe, E.: Adaptive resilience of the cognitive packet network in the presence of network worms. In: Proceedings of the NATO Symposium on C3I for Crisis, Emergency and Consequence Management, pp. 11–12 (2009)
38. Sakellari, G., Gelenbe, E.: Demonstrating cognitive packet network resilience to worm attacks. In: Proceedings of the 17th ACM Conference on Computer and Communications Security, pp. 636–638 (2010)
39. Soe, Y.N., Feng, Y., Santosa, P.I., Hartanto, R., Sakurai, K.: Machine learning-based IoT-botnet attack detection with sequential architecture. Sensors **20**(16), 4372 (2020)
40. Sriram, S., Vinayakumar, R., Alazab, M., Soman, K.: Network flow based IoT botnet attack detection using deep learning. In: IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 189–194. IEEE (2020)
41. Statt, N.: How an army of vulnerable gadgets took down the web today, October 2016. https://www.theverge.com/2016/10/21/13362354/dyn-dns-ddos-attack-cause-outage-status-explained
42. Tuan, T.A., Long, H.V., Kumar, R., Priyadarshini, I., Son, N.T.K., et al.: Performance evaluation of botnet DDOS attack detection using machine learning. Evol. Intell., 1–12 (2019)
43. Tzagkarakis, C., Petroulakis, N., Ioannidis, S.: Botnet attack detection at the IoT edge based on sparse representation. In: 2019 Global IoT Summit (GIoTS), pp. 1–6. IEEE (2019)

Optimizing Energy Usage for an Electric Drone

Tadeusz Czachórski[1](✉), Erol Gelenbe[1,3], Godlove Suila Kuaban[1],
and Dariusz Marek[2]

[1] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,
Baltycka 5, 44–100 Gliwice, Poland
{tadek,seg,gskuaban}@iitis.pl

[2] Silesian University of Technology, Akademicka 16, 44–100 Gliwice, Poland
dariusz.marek@polsl.pl

[3] Laboratoire I3S CNRS, Université Côte d'Azur, 06100 Nice, France

Abstract. Unmanned Aerial Vehicles (UAVs) are rapidly gaining popularity in a wide variety of applications, e.g., agriculture, health care, environmental management, supply chains, law enforcement, surveillance, and photography. Dones are often powered by batteries, making energy a critical resource that must be optimised during the mission of the drone. The duration of a done's mission depends on the amount of energy required to perform some manoeuvering actions (takeoff, level flight, hovering, and landing), the energy required to power the ICT modules in the drone, the drone's speed, payload, and the wind. In this paper, we present a model that minimizes the energy consumption of a low power drone and maximizes the time required to completely drain the drone's battery and ensure the safe landing of the drone.

Keywords: Drones · Battery capacity · Diffusion approximation · Mission optimization · Energy

1 Introduction

The recent advances in Unmanned Arial Vehicle (UAV) technologies (e.g., data collection, data storage, data processing, data transmission, data security, delivery of loads) [32] have increased their adoption rate for military and commercial applications. The fast adoption rate is partly driven by the decrease in the cost of drones and granting licenses to commercial service providers and hobbyists. Some of the industries that are being transformed through the application of drones include agriculture, environmental management, supply chains, law enforcement, surveillance, and photography [27–29]. At the beginning of the COVID-19 pandemic, drones were used for deliveries [11] and to enforce restriction rules (social distancing, no mass gatherings in open public spaces) designed to slow down the transmission of the virus.

Most drones are powered by batteries, making energy a critical resource that must be optimised during the mission of the drone. One of the responsibility

E. Gelenbe et al. (Eds.): EuroCybersec 2021, CCIS 1596, pp. 61–75, 2022.
https://doi.org/10.1007/978-3-031-09357-9_6

of a drone pilot is to ensure that the drone returns with enough energy in the battery that is sufficient for safe landing after its mission. If the drone's battery is completely depleted during its mission, it will crash to the ground and could damage the drone or result in a lawsuit if it damages properties or causes harm to human life. Even the most experienced drone "ground pilots" sometimes encounter drone crashes due to battery depletion. It is difficult to estimate how much time is required to completely deplete the energy stored in the battery during flight because a complex interaction of multiple factors influences the battery energy depletion process in drones. These factors include weather (e.g., wind, temperature), drone speed, the ICT-related functionalities performed by the drone, and the weight of the drone and the load carried by the drone (if any). The energy stored in the battery could also be rapidly depleted due to cyber attacks, which are designed to induce the ICT systems of the drone to draw more energy from the drone unnecessarily. Some drones are configured to return to the operator at predefined battery levels and to land at 15% battery level automatically. Therefore, the drone operator should ensure the safe landing of the drone while preventing any harm to human lives.

To adapt a UAV to perform its functionalities for a given application, advanced on-board information and communication technology significantly increase its energy needs during a mission [6] because of the computationally intensive visual information processing before transmission or storage [9]. Using multiple cooperating UAVs to conduct a mission [23] also increases the computational burden and energy consumption of each UAV, in order to coordinate movements and create a consistent view of the events or scenes that are monitored [21], also leading to additional on-board energy consumption from communications [22], and more on-board software [31]. On-board computing and communication equipment cannot easily be put to sleep to save energy, to avoid compromising the real-time needs which would be impaired by "wake-up" delays [17].

Since careful usage of the UAVs energy budget is needed to achieve the best possible mission output from the battery storage and possible other on-board energy sources such as phtovoltaic and fuel cells, the optimization of the power consumption of an UAV via its speed was studied in [5,7,34]. More broadly, energy consumption is also a major concern in information processing systems [31] and it has been analyzed via a variety of models with the purpose of understanding and minimizing the energy consumption in this area in general [16,22].

However, the energy used to perform functions such as encryption, compression of multimedia data, and communications is significant. In addition, the interplay of multiple factors influencing energy consumption implies that the energy drawn from the battery is not deterministic. Any energy harvesting mechanism that is used on-board is also influenced by the environment. Therefore, both the energy generation and consumption processes on-board a UAV need to be modelled as stochastic processes.

Markovian stochastic models have been applied to model the changes in the energy content of a battery [4,10,18,19,24–26,33]. However, the Poisson assumption in the arrival of energy packets into the battery [15] may deviate from reality. This is why we apply a diffusion model [12,14,20,30] where the interarrival times of energy and the depletion times of the energy follow any distribution, as proposed in earlier work on energy consumption and battery models [1,2,13].

In this paper, we present an optimization model to minimize the energy consumption of a low power drone, and hence maximize the time required to completely drain the drone's battery and ensure the safe landing of the drone. The rest of the paper is organised as follows: Sect. 2 contains a diffusion model of a drone's battery, Sect. 3 contains the proposed optimization model, we present some numerical examples in Sect. 4 and then conclude in Sect. 5.

2 Diffusion Process for the Energy Depletion Process of the Drone

The amount of energy present in the battery at time t may be represented by a diffusion process. This process is frequently used to approximate more complex and analytically intractable stochastic process. It is a strong Markov process with continuous time and continuous space (continuous sample path). We demonstrate how it may be used to evaluate the time after which a device consumes a fixed amount of energy if the consumption per time unit is random.

Consider a Wiener (diffusion) process $X(t)$, corresponding to the energy stored at a battery at the time t. Its changes at unit time have mean β and variance α. For simplicity, we assume that β, α are constant. They can be interpreted as instantaneous mean and variance of the change of $X(t)$

$$\beta = \lim_{\Delta t \to 0} \frac{E[X(t+\Delta t) - X(t)]}{\Delta t}$$
$$\alpha = \lim_{\Delta t \to 0} \frac{Var[X(t+\Delta t) - X(t)]}{\Delta t}.$$

The process' probability density function (pdf) $f(x,t;x_0)$

$$f(x,t;x_0)dx = P[x \le X(t) < x + dx \mid X(0) = x_0]$$

is defined by the diffusion equation (parabolic partial differential equation), e.g. [8]

$$\frac{\partial f(x,t;x_0)}{\partial t} = \frac{\alpha}{2}\frac{\partial^2 f(x,t;x_0)}{\partial x^2} - \beta \frac{\partial f(x,t;x_0)}{\partial x}. \quad (1)$$

For the unrestricted process starting from the point x_0

$$f(x,t;x_0) = \frac{1}{\sqrt{2\Pi\alpha t}} \exp(\frac{(x - x_0 - \beta t)^2}{2\alpha t}) \quad (2)$$

and the incremental changes of $X(t)$ at interval dt

$$dX(t) = X(t + dt) - X(t)$$

are normally distributed with the mean βdt and variance αdt.

If the value of the diffusion process represents the energy content of the battery, then the life time the battery is corresponds to the time the diffusion process needs to pass from the initial point $x_0 = B > 0$, where B is the maximum volume of the battery to $x = 0$. If we refer it to the UAV mission, it corresponds to its maximal duration.

The distribution of the amount of energy present in the battery at time t is given by the Eq. (1) with the absorbing barrier at $x = 0$, i.e. the process is ended when it comes to zero. It corresponds to the condition

$$\lim_{x \to 0} f(x, t; x_0) = 0.$$

The problem of diffusion with absorbing barrier was studied e.g. in [8] and the solution in Eq. (3) was obtained with the use of the method of images: on may treat the barrier as a mirror, and the solution is a superposition of two unrestricted processes, one of unit strength, starting at the origin, and the other of strength $-\exp(\frac{2\beta x_0}{\alpha})$ starting at $x = 2x_0$. It yields

$$\begin{aligned} f(x, t; x_0) = \frac{1}{\sqrt{2\Pi\alpha t}} & \Big[\exp(-\frac{(x-\beta t)^2}{2\alpha t}) \\ & - \exp(\frac{2\beta x_0}{\alpha} - \frac{(x - 2x_0 - \beta t)^2}{2\alpha t})\Big] \end{aligned} \tag{3}$$

The pdf of the first passage time distribution for a diffusion process that starts from the point $x = x_0$ and is absorbed at $x = 0$ is

$$\begin{aligned} \gamma_{x_0,0}(t) &= \int_{0+}^{\infty} \frac{\partial f(x, t; x_0)}{\partial t} dx \\ &= \int_{0+}^{\infty} [\frac{\alpha}{2} \frac{\partial^2 f(x, t; x_0)}{\partial x^2} - \beta \frac{\partial f(x, t; x_0)}{\partial x}] dx \\ &= \lim_{x \to 0} [\frac{\alpha}{2} \frac{\partial f(x, t; x_0)}{\partial x} - \beta f(x, t; x_0)] \\ &= \frac{x_0}{\sqrt{2\Pi\alpha t^3}} e^{-\frac{(x_0 - \beta t)^2}{2\alpha t}} , \end{aligned} \tag{4}$$

with the Laplace transform

$$\bar{\gamma}_{x_0,0}(s) = e^{-x_0 \frac{\beta + \sqrt{\beta^2 + 2\alpha s}}{\alpha}} . \tag{5}$$

Equation (4) presents a probability density function in case of $\beta < 0$, when probability that the process will reach the barrier equals 1, and $\int_0^\infty \gamma_{x_0,0}(t) dt = 1$. Otherwise, for $\beta > 1$, the probability that the process ends at the barrier is $e^{-2\beta x_0/\alpha}$ and the conditional pdf is $\gamma'_{x_0,0}(t) = \gamma_{x_0,0}(t) e^{2\beta x_0/\alpha}$ and $\bar{\gamma}'_{x_0,0}(s) =$

$\bar{\gamma}_{x_0,0}(s)e^{2\beta x_0/\alpha}$. The same refers to the case $\beta < 0$ with the initial point x_0 placed left to the absorbing barrier.

From (4) or (5) we compute the moments of the time the battery is active

$$E[\gamma_{x_0,0}] = \frac{x_0}{|\beta|}, \qquad E[\gamma^2_{x_0,0}] = \frac{|\beta|x_0^2 + \alpha x_0}{|\beta|^3}.$$

In battery model, assuming $x_0 = B$, the pdf given by Eq. (3) determines the distribution of the energy still in the battery, and Eq. (4) the battery life time distribution. Let us imagine that the units of energy are consumed with the mean rate μ units and σ_B^2 is the variance of time intervals between consumption of energy units. It means that the number of consumed energy units in time Δ has the distribution close to normal with mean $\mu\Delta$ and variance $\sigma_B^2\mu^3\Delta$ and the parameters of the diffusion process are $\beta = -\mu$ and $\alpha = \sigma_B^2\mu^3 = C_B^2\mu$, where $C_B^2 = \sigma_B^2\mu^2$ is the squared coefficient of variation (variance/mean^2) of this distribution. A numerical example illustrating the pdf of the first passage time distribution is given in Fig. 1.

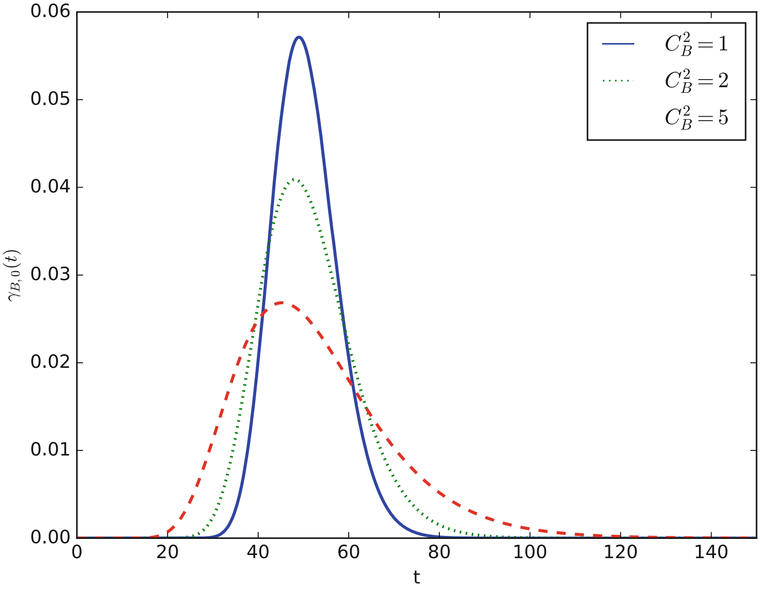


Fig. 1. Probability density function $\gamma_{B,0}(t)$ of the first passage time from a full battery at $B = 50$ to an empty battery at $x = 0$, i.e. the pdf of the battery life time distribution, when the mean power consumption (energy consumption per time unit) is $\mu = 1$. The curves show the influence of the squared coefficient of variation of energy consumption per time unit $C_B^2 = \sigma_B^2\mu^2$ on the life time. We see how the increase of C_B^2 increases the variance of the battery life time distribution

3 Energy Optimization for an UAV During Its Mission

We investigate a problem of UAV control where the energy is limited by the volume of the battery supplying energy to the UAV. The control should maximise a chosen reward function.

There are two phases of the UAV mission. During the first one, the UAV uses all its functions, including the transmission of the collected images. When the energy goes below a certain level of b, the UAV passes to the second, energy-saving phase before landing. During this phase, it is still collecting images, but they are not transmitted. The diffusion parameters, corresponding to energy consumption, are different in both phases, see Fig. 2.

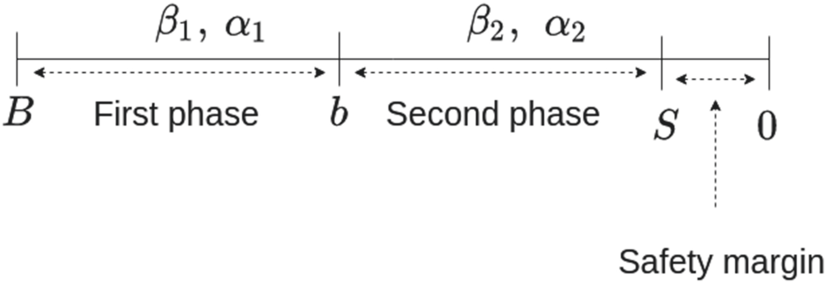


Fig. 2. The two phases of the UAV mission: in the first phase we assume that energy is used for flying, for ground communications and data acquisition. The second phase focuses on the landing phase, including any indispensable data acquisition and communications for the return to the landing base.

The data recorded during the second phase may be accessed only after landing when their validity has partially deteriorated. We assume in the reward function (7) that the value (relevance) of these data is decreasing following a certain function $\Theta(\cdot)$ with the time elapsed between their acquisition and availability.

The formal description of the problem is as follows.

Let $B > 0$ denotes the battery capacity before the UAV platform starts its mission, and S be a minimum value of energy that the UAV battery must contain when it lands after ending the mission, with $B > S \geq 0$.

Let $u,\ v$ be non-negative real numbers such that $B \geq u > v \geq S$.

Define the non-negative random variables τ_u and $Y_{\tau_u}[u,v]$ such that the diffusion process $D \equiv \{X_t, t \geq 0\}$ with $X_0 = B$ has the values

$$\tau_u = \{\inf t : X_t = u\}, \tau_u + Y_{\tau_u}[u,v]\} = \{\inf t > u : X_t = v\}. \qquad (6)$$

Thus τ_u is the first passage time of the diffusion D to level u. Also $\tau_u + Y_{\tau_u}[u,v]$ is the first passage time of D to level v at time $\tau_u + Y_{\tau_u}[u,v]$ after its first passage time to u at time τ_u.

Then our problem is to choose a decision point represented by an “energy switching level” b, $B \geq b \geq S$ for the battery, from “normal energy consumption” to “reduced energy consumption”, which maximises the useful duration of the mission.

Thus we must solve the following maximization problem:

$$\max_{b \in [S,B]} \{C = \ E[\tau_b] + E[\int_0^{Y_\tau[b,S]} dt \Theta(Y_\tau[b,S] - t]\} \ . \tag{7}$$

The first phase’s duration corresponds to the first passage time from B to b, and the second phase is the first passage time from b to S.

Denote by μ_i the mean intensity of the power consumption per time unit at phase i, $i = 1, 2$ (i.e. $1/\mu_i$ is the mean consumption per time unit) and α_i its variance, representing diffusion parameters.

The densities of the duration of the phases are

$$f_1(t) = \gamma_{B,b}(t) = \frac{B-b}{\sqrt{2\Pi\alpha_1 t^3}} \exp(-\frac{(B-b-\mu_1 t)^2}{2\alpha_1 t})$$

$$f_2(t) = \gamma_{b,S}(t) = \frac{b-S}{\sqrt{2\Pi\alpha_2 t^3}} \exp(-\frac{(b-S-\mu_2 t)^2}{2\alpha_2 t})$$

and the mean time of the first phase is

$$E_1 = \int_0^\infty t f_1(t) dt = \frac{(B-b)}{\mu_1}.$$

The reward function C becomes

$$C = E_1 + \int_0^\infty y f_2(y) \int_0^y \Theta(y-t) dt \, dy \tag{8}$$

and we are looking for b, which maximises

$$C = \frac{(B-b)}{\mu_1} + \int_0^\infty y \frac{b-S}{\sqrt{2\Pi\alpha_2 y^3}} \exp(-\frac{(B-S-\mu_2 y)^2}{2\alpha_2 y}) \int_0^y \Theta(y-t) dt \, dy. \tag{9}$$

In the numerical examples below we assume exponential and linear Θ function

$$\Theta(y-t) = e^{-a(y-t)}, \tag{10}$$

$$\text{or} \qquad \Theta(y-t) = \begin{cases} 1 - a(y-t) & \text{for} \quad y \leq 1/a - t \\ 0 & \text{for} \quad y \geq 1/a - t; \end{cases} \tag{11}$$

$\Theta(y-t) = 0$ means that information older than $1/a$ is useless.

4 Numerical Example

We assume that the battery capacity is $B = 50$ energy units, safety margin is $S = 5$ energy units, and $\mu_1 = 0.2$, μ_2 takes several values $\mu_2 = 0.05, 0.1, 0.12, 0.15$, and $\alpha_1 = \alpha_2 = 1$.

A few numerical results giving $C(b)$ for various parameters are displayed in Figs. 3, 4, 5, 6, 7, 8 and 9. In general, they demonstrate the sensibility of $C(b)$ on its parameters and demonstrate the important differences introduced by the deterioration function type. In some cases, the maximum of $C(b)$ is inside the interval $[0, B - S]$, sometimes the function is monotonic, and the maximum is on the edge of the interval. Maximum of $C(b)$ at $b = 50$ means that only the second phase is recommended; maximum at $b = 5$ means we should have only the first phase. It depends on the ratio of the speed of energy consumption in both phases and the function lowering the value of delayed results. A few numerical results giving $C(b)$ for various parameters are displayed in Figs. 3–9. In general, they demonstrate the sensibility of $C(b)$ on its parameters and demonstrate the notable differences introduced by the deterioration function type. In some cases, the maximum of $C(b)$ is inside the interval $[0, B - S]$, sometimes the function is monotonic, and the maximum is on the edge of the interval. Maximum of C(b) at $b = 50$ means that only the second phase is recommended; maximum at $b = 5$ means we should have only the first phase. It depends on the ratio of the speed of energy consumption in both phases and the function lowering the value of

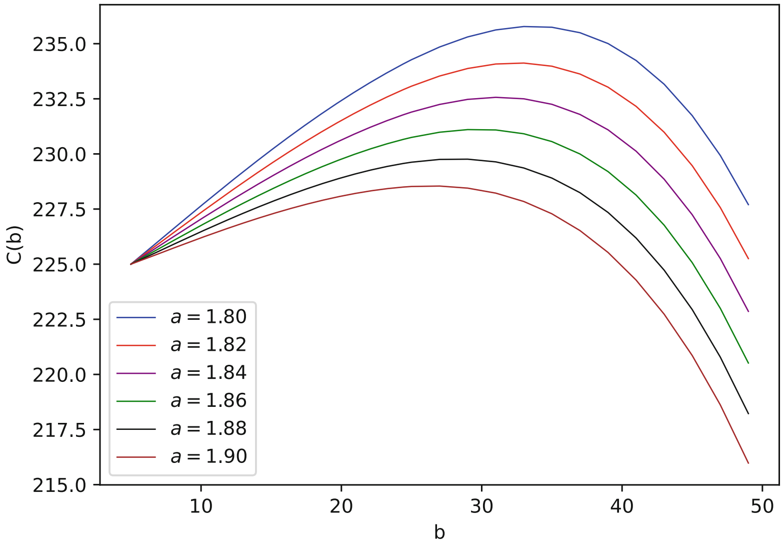


Fig. 3. The reward function $C(b)$ defined by Eq. (9) and to be maximized; b is the energy level switching the performance of UAV from normal mode to energy saving mode, for exponential function Θ defining the decrease with time of the value of previously gathered data, see Eq. (10), with parameters $\mu_2 = 0.1$, $a \in [1.8, 1.9]$

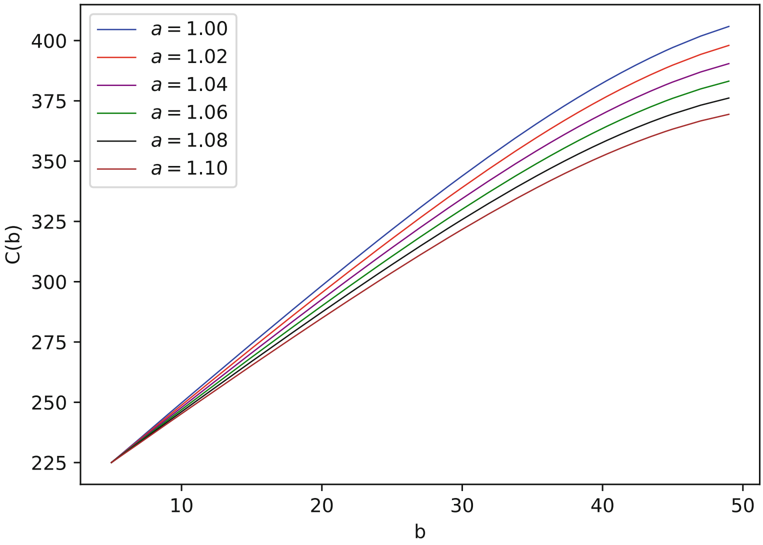


Fig. 4. The reward function $C(b)$ defined by Eq. (9) and to be maximized; b is the energy level switching the performance of UAV from normal mode to energy saving mode, for exponential function Θ defining the decrease with time of the value of previously gathered data, see Eq. (10), with parameters $\mu_2 = 0.1$, $a \in [1.0, 1.1]$

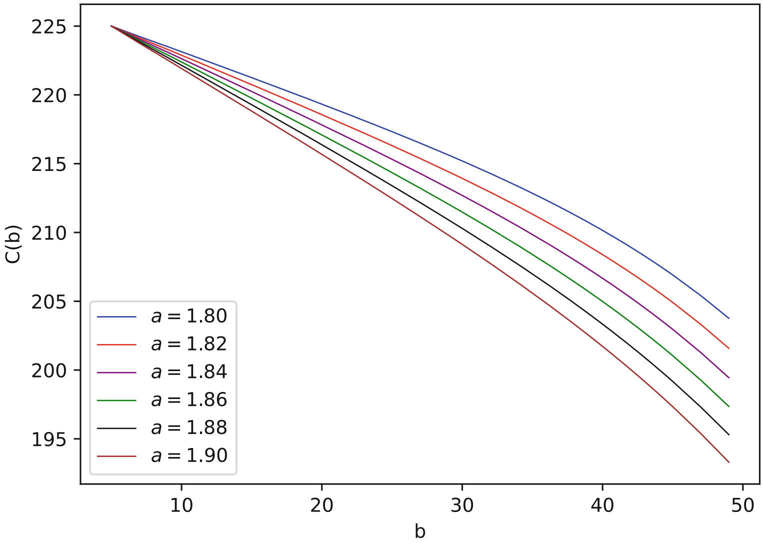


Fig. 5. The reward function $C(b)$ defined by Eq. (9) and to be maximized; b is the energy level switching the performance of UAV from normal mode to energy saving mode, for exponential function Θ defining the decrease with time of the value of previously gathered data, see Eq. (10), with parameters $\mu_2 = 0.12$, $a \in [1.8, 1.9]$

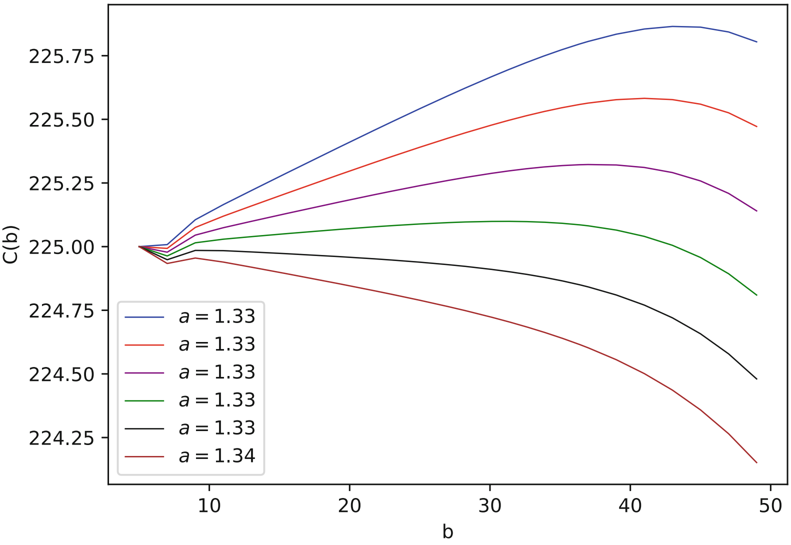


Fig. 6. The reward function $C(b)$ defined by Eq. (9) and to be maximized; b is the energy level switching the performance of UAV from normal mode to energy saving mode, for exponential function Θ defining the decrease with time of the value of previously gathered data, see Eq. (10), with parameters $\mu_2 = 0.15$, $a \in [1.326, 1.336]$

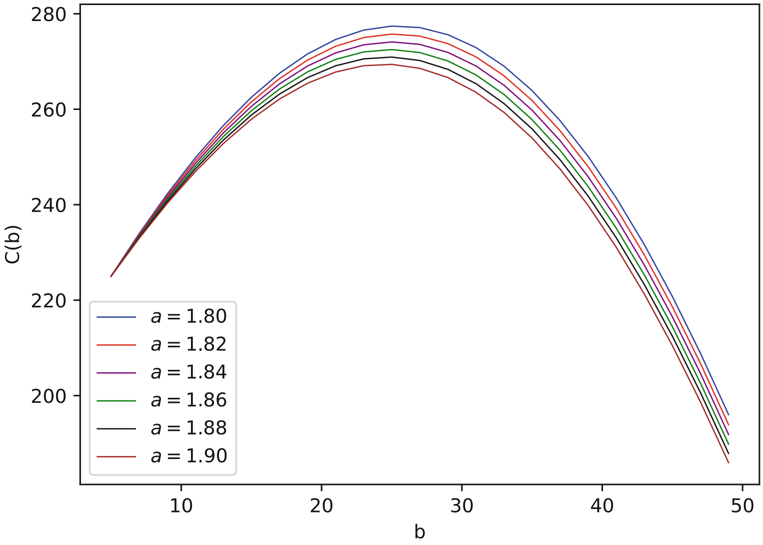


Fig. 7. The reward function $C(b)$ defined by Eq. (9) and to be maximized; b is the energy level switching the performance of UAV from normal mode to energy saving mode, for exponential function Θ defining the decrease with time of the value of previously gathered data, see Eq. (10), with parameters $\mu_2 = 0.05$, $a \in [1.8, 1.9]$

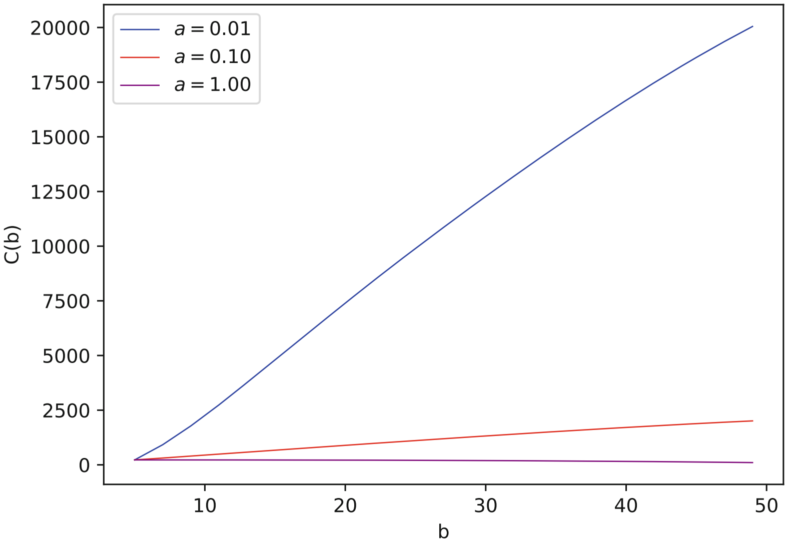


Fig. 8. The reward function $C(b)$ defined by Eq. (9) and to be maximized; b is the energy level switching the performance of UAV from normal mode to energy saving mode, for linear function Θ defining the decrease with time of the value of previously gathered data, see Eq. (11), with parameters $\mu_2 = 0.10$, $a = 0.01, 0.1, 1000$

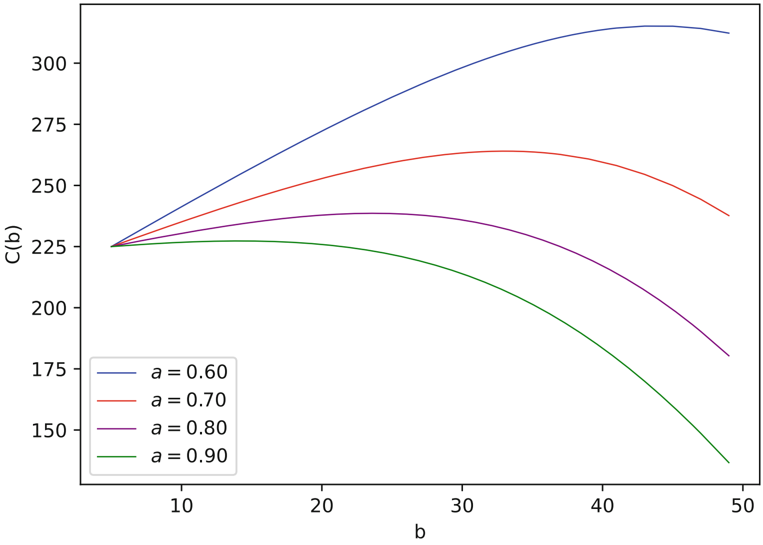


Fig. 9. The reward function $C(b)$ defined by Eq. (9) and to be maximized; b is the energy level switching the performance of UAV from normal mode to energy saving mode, for linear function Θ defining the decrease with time of the value of previously gathered data, see Eq. (11), with parameters $\mu_2 = 0.10$, $a \in [0.6, 0.9]$.

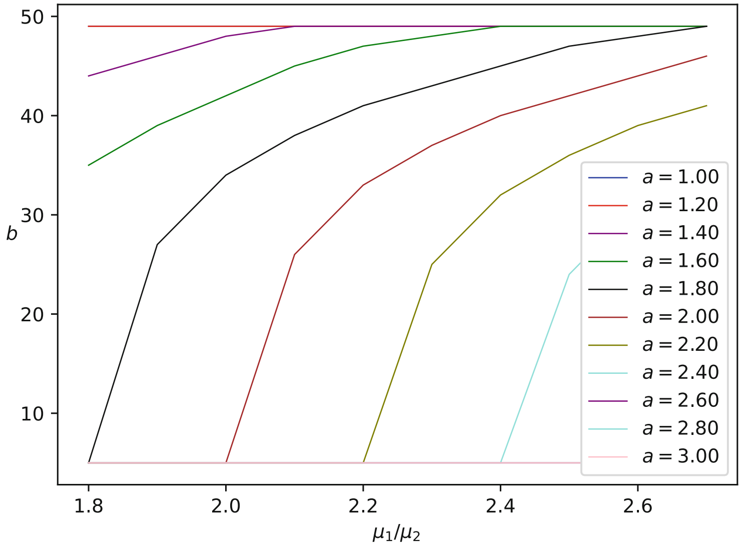


Fig. 10. The value of b, i.e. the energy level switching the performance of UAV from normal mode to energy saving mode, maximizing the reward function $C(b)$ defined by Eq. (9) plotted as a function of μ_1/μ_2, where μ_1 is the speed of energy consumption in normal mode and μ_2 is the speed of energy consumption in energy saving mode, for fixed $\mu_1 = 0.2$.

delayed results. We see it in a more general way in Fig. 10 where the values of b giving the maximum of C are plotted as the function of μ_1/μ_2 ($\mu_1 = 0.2$).

5 Conclusions

The duration of a done's mission depends on the amount of energy required to perform some manoeuvering actions (takeoff, level flight, hovering, and landing) [3] and the energy required to power the ICT modules in the drone. The energy required to drive the drone depends on the manoeuvering action taken, the drone's speed, payload, and the wind. Although the amount of energy required to drive the drone is often far greater than the energy required to power the ICT modules, the influence of ICT energy consumption on the duration of the drone's mission could become significant (especially for drones that draw small amount of energy for flight but perform complex ICT functionalities). Also, cyber security attacks designed to increase the amount of transmission or computations executed by the drone and deplete its battery faster could rapidly deplete the energy stored in the battery.

Since for some set of parameters, the reward function $c(b)$ has a maximum for $b \in [0, B - S]$, the by devising strategies to reduce the energy consumption, when the energy in the battery reaches a define threshold level b, increases the chances that it will complete its mission and land safely. Therefore, the decision point to transition from the normal phase to the energy saving phase to be chosen in such as way as to minimise the energy consumption and maximise the battery lifetime.

Acknowledgements. The work presented in this paper was partially supported by the Research and Innovation project: "Security by Design IoT Development and Certificate Framework with Frontend Access Control (IOTAC)," that is funded by the European Commission under the H2020-SU-ICT-2018-2020/H2020-ICT-2019 Program through Grant Agreement 952684.

References

1. Abdelrahman, O., Gelenbe, E.: A diffusion model for energy harvesting sensor nodes. In: Proceedings of 24th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 154–158. IEEE (2016)
2. Abdelrahman, O.H., Gelenbe, E.: Packet delay and energy consumption in non-homogeneous networks. Comput. J. **55**(8), 950–964 (2012)
3. Abeywickrama, H.V., Jayawickrama, B.A., He, Y., Dutkiewicz, E.: Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance. IEEE Access **6**, 58383–58394 (2018)
4. Arabi, S., Sabir, E., Elbiaze, H., Sadik, M.: Data gathering and energy transfer dilemma in UAV-assisted flying access network for IoT. Sensors **18**(1519), 1–20 (2018)
5. Baek, D., Chen, Y., Poncino, M.: Battery-aware energy model of drone delivery tasks. In: Proceedings of the International Symposium on Low Power Electronics and Design, pp. 1–6. ACM (2018). https://doi.org/10.1145/3218603.3218614
6. Cai, L.X., Poor, H.V., Liu, Y., Luan, T.H., Shen, X., Mark, J.W.: Dimensioning network deployment and resource management in green mesh networks. IEEE Wirel. Commun. **18**, 58–65 (2011)
7. Chen, Y., Baek, D., Bocca, A., Macii, A., Macii, E., Poncino, M.: A case for a battery-aware model of drone energy consumption. In: Proceedings of the 2018 IEEE International Telecommunications Energy Conference (INTELEC), pp. 1–8. IEEE (2018). https://doi.org/10.1109/INTLEC.2018.8612333
8. Cox, R.P., Miller, H.D.: The Theory of Stochastic Processes. Chapman and Hall, London (1965)
9. Cramer, C.E., Gelenbe, E.: Video quality and traffic QoS in learning-based subsampled and receiver-interpolated video sequences. IEEE J. Sel. Areas Commun. **18**(2), 150–167 (2000)
10. De Cuypere, E., De Turck, K., Fiems, D.: A queueing model of an energy harvesting sensor node with data buffering. Telecommun. Syst. **67**(2), 281–295 (2017). https://doi.org/10.1007/s11235-017-0338-8
11. Euchi, J.: Do drones have a realistic place in a pandemic fight for delivering medical supplies in healthcare systems problems? Chin. J. Aeronaut. **34**, 182–190 (2020)
12. Gelenbe, E.: Probabilistic models of computer systems, Part II: diffusion approximations, waiting times and batch arrivals. Acta Informatica **12**(4), 285–303 (1979)
13. Gelenbe, E.: A diffusion model for packet travel time in a random multihop medium. ACM Trans. Sen. Netw. **3**(2), 10-es (2007). https://doi.org/10.1145/1240226.1240230

14. Gelenbe, E.: Search in unknown random environments. Phys. Rev. E **82**(6), 061112 (2010)
15. Gelenbe, E.: Energy packet networks: ICT based energy allocation and storage. In: Rodrigues, J.J.P.C., Zhou, L., Chen, M., Kailas, A. (eds.) GreeNets 2011. LNICST, vol. 51, pp. 186–195. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33368-2_16
16. Gelenbe, E.: Synchronising energy harvesting and data packets in a wireless sensor. Energies **8**(1), 356–369 (2015)
17. Gelenbe, E., Iasnogorodski, R.: A queue with server of walking type (autonomous service). In: Annales de l'IHP Probabilités et statistiques, vol. 16, pp. 63–73 (1980)
18. Gelenbe, E., Kadioglu, Y.M.: Battery attacks on sensors. In: International Symposium on Computer and Information Sciences, Security Workshop (2018)
19. Gelenbe, E., Kadioglu, Y.M.: Energy life-time of wireless nodes with network attacks and mitigation. In: 2018 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 1–6. IEEE (2018)
20. Gelenbe, E., Muntz, R.R.: Probabilistic models of computer systems. Part I: exact results. Acta Informatica **7**(1), 35–60 (1976)
21. Gelenbe, E., Sevcik, K.: Analysis of update synchronization for multiple copy databases. In: 3rd Berkeley Workshop on Distributed Data and Computer Networks, pp. 69–90 (1978)
22. Gelenbe, E., Silvestri, S.: Reducing power consumption in wired networks. In: 2009 24th International Symposium on Computer and Information Sciences, pp. 292–297. IEEE (2009)
23. Hu, J., Lanzon, A.: An innovative tri-rotor drone and associated distributed aerial drone swarm control. Rob. Auton. Syst. **103**, 162–174 (2018) https://doi.org/10.1016/j.robot.2018.02.019. https://www.sciencedirect.com/science/article/pii/S0921889017308163
24. Kadioglu, Y.M., Gelenbe, E.: Packet transmission with k energy packets in an energy harvesting sensor. In: Proceedings of the 2nd International Workshop on Energy-Aware Simulation, pp. 1–6 (2016)
25. Kadioglu, Y.M., Gelenbe, E.: Wireless sensor with data and energy packets. In: 2017 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 564–569. IEEE (2017)
26. Kadioglu, Y.M., Gelenbe, E.: Product-form solution for cascade networks with intermittent energy. IEEE Syst. J. **13**(1), 918–927 (2018)
27. Koparan, C., Koc, A.B., Privette, C.V., Sawyer, C.B.: In situ water quality measurements using an unmanned aerial vehicle (UAV) system. Water **10**(3), 264 (2018). https://doi.org/10.3390/w10030264
28. Koparan, C., Koc, A.B., Privette, C.V., Sawyer, C.B.: Autonomous in situ measurements of noncontaminant water quality indicators and sample collection with a UAV. Water **11**(3), 604 (2019). https://doi.org/10.3390/w11030604
29. Koparan, C., Koc, A.B., Privette, C.V., Sawyer, C.B.: Adaptive water sampling device for aerial robots. Drones **4**(1), 5 (2020). https://doi.org/10.3390/drones4010005
30. Marin, G.A., Mang, X., Gelenbe, E., Onvural, R.O.: Statistical call admission control. IEEE Commun. Lett. **222**(824) (2001)
31. Pernici, B., Aiello, M., Vom Brocke, J., Donnellan, B., Gelenbe, E., Kretsis, M.: What is can do for environmental sustainability: a report from CAISE'11 panel on green and sustainable is. Commun. Assoc. Inf. Syst. **30**(1), 18 (2012)

32. Sharma, A., Basnayaka, C.M., Jayakody, D.N.K.: Communication and networking technologies for UAVs: a survey. J. Netw. Comput. Appl. **168**, 102739 (2020). https://doi.org/10.1016/j.jnca.2020.102739.S2CID221507920
33. Sharma, V., Rajesh, R.: Queuing theoretic and information theoretic capacity of energy harvesting sensor nodes. In: Proceedings of 2011 Conference Record of the Forty Fifth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), pp. 383–388. IEEE, Pacific Grove (2011). https://doi.org/10.1109/ACSSC.2011.6190024
34. Thibbotuwawa, A., Nielsen, P., Zbigniew, B., Bocewicz, G.: Energy consumption in unmanned aerial vehicles: a review of energy consumption models and their relation to the UAV routing. In: Świątek, J., Borzemski, L., Wilimowska, Z. (eds.) ISAT 2018. AISC, vol. 853, pp. 173–184. Springer, Cham (2019). https://doi.org/10.1007/978-3-319-99996-8_16

T-RAID: TEE-based Remote Attestation for IoT Devices

Roland Nagy, Márton Bak, Dorottya Papp, and Levente Buttyán(✉)

Laboratory of Cryptography and System Security (CrySyS Lab),
Department of Networked Systems and Services,
Budapest University of Technology and Economics, Budapest, Hungary
buttyan@crysys.hu
https://www.crysys.hu/

Abstract. The Internet of Things (IoT) consists of network-connected embedded devices that enable a multitude of new applications, but also create new risks. In particular, embedded IoT devices can be infected by malware. Operators of IoT systems not only need malware detection tools, but also scalable methods to reliably and remotely verify malware freedom of their IoT devices. In this paper, we address this problem by proposing T-RAID, a remote attestation scheme for IoT devices that takes advantage of the security guarantees provided by a Trusted Execution Environment running on each device.

Keywords: Internet of things · Embedded systems · Malware · Remote attestation · Trusted Execution Environment

1 Introduction

The Internet of Things (IoT) consists of network-connected embedded devices that enable a multitude of new applications in various domains, such as industrial automation, transportation, building automation, healthcare, and agriculture – just to mention a few of them. The use of IoT technologies can make applications *smarter*: they provide the technological foundations for transforming buildings into smart buildings, cities into smart cities, transportation systems into intelligent transportation systems, healthcare into personalized healthcare, agriculture into precision agriculture, and factories into smart factories.

However, as usual, new technologies also create new risks. In particular, due to the increasing levels of automation and connectedness, our new, smart and intelligent applications are now exposed to cyberattacks. To address the problem, academic researchers and industry alliances are actively working on IoT security solutions [1,12,13], standards [8,10], and guidelines [7,16,17], and regulatory

The presented work was carried out within the SETIT Project (2018-1.2.1-NKP-2018-00004), which has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the 2018-1.2.1-NKP funding scheme.

E. Gelenbe et al. (Eds.): EuroCybersec 2021, CCIS 1596, pp. 76–88, 2022.
https://doi.org/10.1007/978-3-031-09357-9_7

bodies are also making steps[1] to ensure that those solutions, standards, and guidelines are indeed used and followed in practice.

One particular security issue is that IoT devices can be infected by malware [2,4], which can alter their behavior, endangering the integrity and the availability of IoT systems, and undermining the trustworthiness of the smart applications based on them. Hence, system operators need malware detection solutions adapted to the constraints of IoT systems [20]. In addition, they also need scalable methods to reliably verify malware freedom of IoT devices in their systems. In this paper, we address this problem by proposing a remote attestation scheme for IoT devices that takes advantage of the security guarantees provided by a Trusted Execution Environment (TEE) running on the device.

Attestation is meant to be a process whereby a trusted verifier reliably checks the state of an untrusted prover, and remote attestation is when this verification is done remotely via a network. In our case, the prover is (a process running on) an IoT device, and it is untrusted, because the device may be compromised by a malware. The verifier is a trusted remote server operated by the system operator. We use remote attestation to prove the malware-free state of the IoT devices to the system operator. Doing this remotely means that the operator does not need physical access to the devices, allowing for large scale, automated verification of all devices in an entire IoT system.

The structure of the paper is the following: We start by giving a brief overview on different approaches to remote attestation in Sect. 2, serving as a review of relevant related work and also providing technical background for our proposal. Next, we introduce T-RAID, our TEE-based solution to secure remote attestation for IoT devices, by first providing a general overview of it in Sect. 3, and then presenting its protocols in more details in Sects. 4 and 5. Finally, we evaluate T-RAID and discuss its properties in Sect. 6, and conclude the paper in Sect. 7.

2 Approaches to Remote Attestation

There exist three general approaches to attestation: hardware-based, software-based, and hybrid attestation. Hardware-based attestation relies on a secure co-processor (e.g., a TPM chip[2]) that can produce a digitally signed summary of the hardware and software state of the device being verified. The summary is typically a hash computed by progressively combining the hashes of the system components and software modules started during the boot process. The key used to sign the summary is kept in the co-processor and protected by its logical access control and physical tamper resistance features. As this approach requires a co-processor, it is typically considered to be too expensive for embedded IoT devices.

[1] The California IoT cybersecurity law SB-327 became effective Jan 1, 2020.

[2] https://trustedcomputinggroup.org/work-groups/trusted-platform-module/ Last visited: Sep 12, 2021.

In contrast to the hardware-based approach, software-based attestation does not require any additional hardware in the prover device. Solutions following this approach (e.g., [18,19]) are typically based on executing a protocol in which the verifier probes the prover and the response of the prover to the probe convinces the verifier that the prover is in a malware-free state. To generate the response, the prover runs a checksum function, which traverses memory locations in a pseudo-random manner (seeded by the verifier's probe). The verifier checks the correctness of the prover's response by computing the same checksum function on the expected memory content of the device. If malware is hiding in the memory, either the checksum of the prover will differ from that computed by the verifier, or the response time of the prover will be longer than expected, as the malware needs to check and redirect memory accesses that refer to locations holding the malware code itself. So besides checking the correctness of the checksum, in this case, the verifier also checks the response time of the prover.

The main problem of software-based attestation is that, in practice, it does not really work over a network due to network jitter, which makes it practically impossible to remotely measure the exact checksum computation time of the prover [14]. Another problem is that a compromised prover can actually delegate checksum computation to a much faster attacker device, which cannot be detected by a remote verifier. In addition, even if we do not consider such delegation attacks, this approach assumes that the checksum computation on the prover cannot be performed faster than the speed of the actual implemention of the checksum function. Unfortunately, this assumption does not always hold [3], leading to attacks where a tricky faster way of computing the checksum leaves time for the malware to check and redirect memory references that would reveal its presence.

Given all these problems, hybrid approaches were proposed (e.g., [5,6]) that are largely software-based, but also require minimal hardware support. For example, in [6], the authors propose a scheme, applicable to attestation purposes, that uses a ROM-based checksum routine and relies on a secret key for authenticating the computed checksum that is kept in memory accessible only by ROM-based code. This latter property is provided by a hardware-based memory access logic, which verifies that the instruction pointer is in the ROM region when the secret key is being accessed. This actually ensures that the confidentiality of the key is preserved, even if the device is infected by a malware. In addition, as the ROM code cannot be modified, integrity of the checksum computation is also ensured. This means that a response authenticated by the secret key must be genuine, and such a response can be verified by a remote verifier.

At this point, a natural question could be the following: What is the minimum hardware support needed for a hybrid remote attestation solution to be secure? This question is investigated in [9], where the authors conclude that the following set of requirements is sufficient and necessary (hence minimal) for secure remote attestation of embedded devices:

1. Custom hardware to enforce exclusive access to a secret key;
2. Reliable and secure memory erasure;

3. Read-only-memory (ROM);
4. Instructions for enabling and atomically disabling interrupts;
5. Custom hardware to enforce that the attestation (checksum) routine can only be invoked by running its first instruction;
6. Secure reset mechanism.

It turns out that the guarantees provided by satisfying the requirements above can also be achieved in another way: In [5], the authors propose a hybrid remote attestation scheme, called HYDRA, that relies on security features provided by a formally verified seL4 microkernel to obtain similar properties. In HYDRA:

1. A privileged process handles the secret key and enforces proper access control to it;
2. Reliable and secure memory erasure is required by [9] to ensure that no information about the secret key is leaked after using it in the checksum authentication. However, the strict memory separation of the seL4 microkernel ensures the same property;
3. ROM is required by [9] to make sure that the checksum routine cannot be modified. Isolated process memory and code integrity checks in seL4 can provide the same property;
4. Prioritized interrupt handling of the microkernel ensures uninterruptable execution of the checksum code that runs with the highest possible priority;
5. Controlled invocation is enforced by operating system support;
6. Secure reset is initiated in [9] whenever an attempt is detected to execute the checksum function from the middle of its code. This is not needed if controlled invocation is enforced.

The authors of [5] claim that using seL4 imposes fewer hardware requirements on the underlying microprocessor, and building upon a formally verified software component increases confidence in security of the overall solution.

3 Overview of T-RAID

Our main idea is to follow the approach of HYDRA [5], but instead of a secure microkernel, we rely on a Trusted Execution Environment (TEE). A TEE provides an isolated environment for trusted processes where they can execute without being interfered by normal, potentially compromised processes. In addition, a TEE also provides secure storage for secrets that is not accessible from outside the TEE. Thus, checking malware-freedom can be implemented in a trusted process running in the TEE and the key used for authenticating the result of the check can be stored in TEE-protected secure memory.

We note that embedded devices equipped with ARM or similarly powerful processors are typically capable of hosting such a TEE. We also note that TEEs usually rely on hardware support to provide their security guarantees.

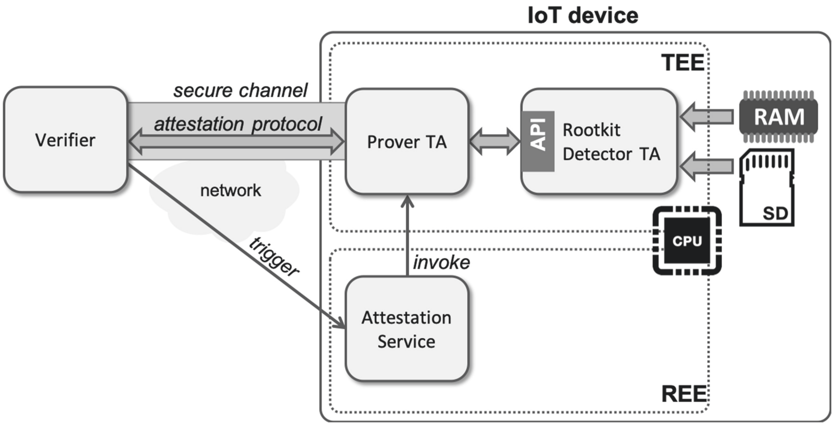


Fig. 1. Overview of the T-RAID architecture.

For example, TrustZone[3] enabled ARM processors support TEEs by offering hardware-enforced memory isolation. A special register in the processor keeps track whether it runs in the so called *Normal World* or in the *Secure World*. In the Normal World, access to certain memory regions and peripherals associated with the Secure World is denied, and this is enforced by the memory bus fabric. On the other hand, processes running in the Secure World have virtually unlimited access to any resources of the embedded device. In addition, switching between Worlds is possible only by invoking a special instruction that guarantees a proper context switch. TEE implementations usually take advantage of this low level support built into the processor itself.

The architecture of our TEE-based remote attestation scheme designed for IoT devices (T-RAID) is illustrated in Fig. 1. In T-RAID, the Verifier is assumed to be a remote entity that interacts with the Prover via a network. The Prover is a trusted application (TA) running in the TEE hosted by the processor of the IoT device. An Attestation Service is running as a normal (untrusted) process in the Rich Execution Environment (REE), also hosted by the processor. The term "rich" refers to the fact that the REE may be provided by a full-blown operating system (OS), such as Linux, that offers an abundance of services to the processes running in the REE. The TEE typically features a much more limited OS that provides only basic services to the trusted applications. Isolation between the TEE and the REE is supported by the processor and its memory management unit.

In order to initiate an attestation session, the Verifier calls the Attestation Service, which is assumed to be always available. If it is not, then this fact already proves that the IoT devices is not in its normal state, and it may be compromised by a malware. The Attestation Service invokes the Prover TA via

[3] https://developer.arm.com/ip-products/security-ip/trustzone Last visited: Sep 12, 2021.

a controlled invocation mechanism provided by the TEE (and supported by the processor). The Prover TA then establishes a secure connection to the Verifier, which is used to execute a remote attestation protocol securely.

In the remote attestation protocol, the Verifier challenges the Prover with a set of tasks. These tasks consists in the execution of certain integrity checks that are implemented by a Rootkit Detector TA, also running in the TEE. The integrity checks requested by the Verifier are invoked by the Prover TA via a well-defined API provided by the Rootkit Detector TA, and the results are sent back to the Verifier via the secure channel.

The Rootkit Detector TA has access to the entire memory of the IoT device and its persistent storage. The memory includes the memory of the processes and the OS kernel running in the REE and the persistent storage includes the file system that they use. The integrity checks implemented by the Rootkit Detector TA analyze this memory and file system, collect relevant data (e.g., the list of processes currently running in the REE), compute hashes (e.g., the hash of the text segment of the OS kernel in the REE), and try to detect anomalies that may signal the presence of a malware (e.g., hooked function pointers).

More information about the remote attestation protocol and the integrity checks of T-RAID is provided below in Sects. 4 and 5, respectively.

We implemented T-RAID as a prototype running in QEMU[4]. The target architecture of our prototype is the ARM processor, and we use OP-TEE[5] as the TEE implementation and Linux as the OS in the REE. Cryptographic functions in our protocols use the mbedTLS cryptographic library[6]. In the sequel, we assume this setup when implementation specific details are described.

4 Remote Attestation Protocol

The remote attestation protocol of T-RAID assumes a secure channel between the Prover TA and the Verifier, thus, a prerequisite for running the protocol is to establish such a secure channel. One can use TLS for this purpose if the TEE implementation supports TLS-protected sockets and the resource constraints of the IoT device permit the use of such a complex protocol as TLS. In our prototpye implementation, we could not use TLS, because OP-TEE does not support TLS-based sockets in the TEE. Instead, we designed and implemented a lightweight secure channel protocol over a raw socket. Our protocol uses an authenticated version of the Diffie-Hellman protocol to establish a shared secret between the Prover TA and the Verifier (where authentication is based on ECDSA signatures); the PBKDF2 key derivation function to derive a 32-byte symmetric encryption key and a 32-byte message authentication key from the shared secret; the AES cipher in CBC mode with PKCS#7 padding to encrypt messages; message sequence numbers to protect against replay attacks;

[4] https://www.qemu.org/ Last visited: Sep 12, 2021.

[5] https://www.op-tee.org/ Last visited: Sep 12, 2021.

[6] https://github.com/ARMmbed/mbedtls Last visited: Sep 12, 2021.

and HMAC with SHA-256 as the hash function to authenticate (encrypted and numbered) messages.

The attestation protocol consists of the exchange of a single attestation request and response. The secure channel guarantees the freshness, integrity, authenticity, and confidentiality of these messages, and most importantly, the Prover's response. The request of the Verifier may contain multiple challenges, each triggering the call of a specific integrity check function of the Rootkit Detector TA. The response of the Prover contains the results of the integrity check functions called. The integrity check functions may return a status code (e.g., 0 for a successful check and 1 for a failure), a hash value (e.g., hash of the REE OS kernel's text segment or recursive hash of some part of the REE file system), or a list of process IDs and process names extracted from various OS kernel data structures (e.g., the process list, the process tree, and the run queues).

The Verifier must be able to verify the results of the checks received in the response of the Prover. For this, we assume that the Verifier stores the hash values expected in correct responses. In case of file system checks, the computed hash value depends on what parts of the file system are actually hashed, therefore, we assume that the Verifier has a mirror of the file system of the IoT device and performs the same hash computation on this mirror to obtain the expected correct hash value. Finally, if the response contains a list of process names, the Verifier can compare that to a pre-defined white list of process names allowed on the IoT device.

5 Integrity Checks

Our integrity checks perform rootkit detection on the IoT device; hence, their successful execution is an assurance of malware freedom. The software component implementing the checks is capable of accessing components of the REE, such as the memory and the file system. The latter is not supported by OP-TEE, so we had to extend and slightly modify the OP-TEE kernel and the `tee-supplicant` daemon (a component of OP-TEE running in the REE). More details on this can be found in our earlier paper [15]. Using the aforementioned capabilities, we implemented numerous checks, each aiming at detecting a different rootkit technique or ensuring the integrity of REE components that our checks rely upon. In this section, we present the integrity checks that the Prover TA can invoke.

5.1 Process Listing

The most important functionality of any kernel is to manage and schedule processes. In order to achieve these goals, the Linux kernel uses so called tasks. A task is approximately equivalent to a thread. Single-threaded processes consist of one task, while multi-threaded processes have one task for each thread, sharing the same address space. These tasks are organized into multiple dynamic data structures. Here we present the ones used by the 5.1 version of the Linux kernel.

Our solution uses these data structures to enumerate processes on the system. Currently, we can list process IDs and names of the processes.

The oldest and simplest data structure in the kernel holding process-related information is the so called process list. This is a doubly-linked circular list of task structures; each task has a `next` and `prev` pointer, pointing to the next and previous tasks in the list, respectively. Using these pointers, we can easily traverse the whole list, starting from the `init_task`; this is the first kernel thread, started at boot.

Another data structure is the process tree. When a process starts another process, it becomes its child, while the new process refers to the original one as its parent. Via this relation, processes form a tree, whose root is the `init_task`. The Linux kernel uses lists to implement this tree. Each process has a pointer to its first and last child, while the children are linked into a doubly-linked circular list. We traverse this data structure recursively in a depth-first manner.

Pid namespaces are used by the kernel as an isolation mechanism. There is an initial pid namespace containing every process. These namespaces use radix trees to account the process IDs in use. These radix trees store `pid` structures with pointers to the tasks using the specific ID. To traverse the initial pid namespace, we implemented a function capable of finding the corresponding `pid` structure in the tree for a given ID.

Finally, we extract process related information from runqueues. These queues are used by the scheduler, and unlike the previous data structures, not every process is accessible from these, only the runnable ones. These are the processes not waiting for anything and not stopped, they can continue their execution, if assigned to a CPU core. Each core has its own runqueue, and runqueues implement data structures for every scheduling policy. For Linux 5.1, this means 3 subschedulers, using lists, red-black trees and nested red-black trees.

5.2 Memory Integrity Checks

We check the integrity of two memory areas of the Linux kernel that are frequently targeted by rootkits: the system call table and the text segment of the kernel itself. System calls are the interface the kernel offers to user-space processes. When processes need to perform actions that are the kernel's responsibility, they invoke the appropriate system call. Such actions include interactions with files, network sockets, etc. The kernel uses an array of function pointers, known as the system call table. Rootkits often replace pointers in this array and re-implement certain system calls. Therefore, we compute the hash of the system call table using the SHA-256 hash algorithm.

Another common and similar rootkit technique is *inline hooking* [11]. In this case, the attacker modifies the code of an existing function, usually by rewriting the first few instructions to a jump such that during execution, the code jumps to the desired replacement. To detect inline hooks, we compute the SHA-256 hash of the entire text segment of the kernel, which contains all the code of the Linux kernel.

5.3 File Integrity Checks

By-default OP-TEE does not provide access to the file systems of the REE, however, it is capable of using it for Secure Storage. The API written for this does not aim to be a general purpose API for file access, so we had to extend it and apply some patches in order to be able to access arbitrary files. Again, for implementation details, the reader is referred to [15].

Our implementation provides a simple interface which can be used to check the integrity of any part of the REE file systems. This can be done invoking two functions, namely `hash_file` and `hash_dir`. The former one expects a filename as parameter, and an output buffer, where the computed hash of the file will be stored. If the file exists, it opens it, reads it by 4096-byte-long blocks, and feeds these blocks into a hash context. After reaching the end of the file, the SHA-256 hash is written to the output buffer. The latter one expects more parameters: a directory name, an output buffer, a boolean indicating if it should hash the directory recursively, and an optional blacklist. In case of non-recursive hashing, all subdirectories will be ignored. If a blacklist is supplied, all elements are checked against it, thereby the hashing files or directories with volatile content can be avoided. The contents of directories are sorted alphabetically.

5.4 Network Checks

We also implemented checks targeting the network stack of the kernel. In this subsystem, rootkits typically implement two kind of attacks: hiding open connections and implementing "magic packet" functionality. In the case of rootkits, this means performing a predefined action, when the infected system receives a specially crafted network packet. So far, we identified one way to hide open sockets and three mechanisms what can be abused by attackers to implement magic packets. For these checks to work properly, we assume that every necessary driver is compiled into the kernel.

The most common way to implement magic packets is using the Netfilter subsystem, the backend of Linux firewall solutions. Netfilter stores firewall rules in so-called chains. Each supported protocol (like IPv4, IPv6, ARP, etc.) has five chains, one for each stage of packet processing. Each chain acts as an arraylist, containing Netfilter hooks, storing function pointers. When a packet is checked against a specific chain, all hooks in the chain are invoked, and the packet is accepted only, if all hooks accept it. These hook functions, however, can have side effects, so an attacker can implement a firewall rule which executes his payload, if certain conditions are met. Our solution traverses all the hooks of every chain, and if any of the function pointers store a value that is not pointing into the text segment of the kernel, it is considered to be a sign of rootkit infection.

We also check structures called `icmp_control`. The kernel uses an array of these to determine what handler function should be executed for different kinds of ICMP packets. The packet's `type` field is used to index this array. We check all function pointers the same way as we did in the case of Netfilter hooks.

The kernel uses `net_protocol` structures to register handler functions for different protocols, like UDP, TCP and IGMP. These structures contain handler, error handler and demultiplexer functions, which can be hooked and used to implement magic packet functionality. We perform the same integrity check on these pointers as described above.

Finally, we implemented a check targeting hidden network connections. Files in the `/proc/net` directory give information about open connections. The content of these files is generated on-demand using `seq_ops`. These objects store function pointers to iterate a specific data structure and display information about its elements. Rootkits often target these to hide open connections, therefore we check these function pointers the same way as we checked the others.

6 Evaluation and Discussion

The presented TEE-based remote attestation scheme, T-RAID, provides security guarantees similar to those of HYDRA [5]:

1. A trusted application in the TEE, the Prover TA, handles the private key used to set up a secure channel with the Verifier. Every message, including the Prover's responses to the Verifier's challenges are authenticated by this channel, which means that the Verifier can be sure that the responses come from the given Prover. In addition, the private key of the Prover is stored in the secure storage of the TEE, hence, the key remains protected and invisible from the potentially compromised REE.
2. Strict separation of the secure memory used by the TEE from processes in the REE prevents the leakage of the private key after it has been used.
3. TEE-based integrity protection of TAs prevents their illegitimate modification by untrusted processes of the REE. This property ensures similar guarantees for TAs as a ROM would ensure. Thus, neither the Prover TA nor the Rootkit Detector TA can be modified, and hence, the integrity checks are performed and their results are reported correctly to the Verifier.
4. Interrupts can be disabled and re-enabled in the TEE. Disabling them when the Rootkit Detector TA is invoked ensures the uninterruptable execution of our integrity checks, with some caveats that we discuss later in this section.
5. TEE-based invocation mechanism of TAs enforces that the execution of a TA always begins at its entry point. This contributes to the correct execution of the integrity checks and correct reporting of their results.
6. Secure reset is not needed, as the TEE enforces the controlled invocation of every TA.

Unfortunately, our current prototype has two known weaknesses. The first one is that file operations in OP-TEE are delegated to the REE side where they can actually be interrupted. Moreover, as file operations usually take a long time, they will almost surely be interrupted by the task scheduler of the OS. This is a limitation of OP-TEE, other TEE implementations may implement file operations within the TEE itself. Nevertheless, if T-RAID is implemented using OP-TEE as the TEE, one has to be aware that property 4 may not hold. Disabling the file system related integrity checks would make property 4 satisfied, but then malware could clear itself from memory and hide its components in persistent storage, from which it may be potentially reloaded later on.

The second weakness is that on multi-core processors, such as most ARM processors, other, potentially untrusted processes may run in parallel to our TAs on different cores. Those processes may interfer with the execution of our integrity checks. For instance, a malware running on a different core could remove a hook from the system call table before it is hashed by our Rootkit Detector TA and put the hook back once the hashing is completed. The only reliable countermeasure to this is disabling all but one cores during the entire attestation process. At the time of this writing, we are experimenting with the implementation of this idea.

7 Conclusions

In this paper, we proposed T-RAID, a TEE-based remote attestation scheme for IoT devices. T-RAID follows the hybrid approach to remote attestation: it is mostly based on software and uses only limited hardware support. Notably, T-RAID relies only on the hardware support provided for the TEE itself by the processor and its memory management unit. Considering that TEEs are already widely supported by certain classes of embedded devices, T-RAID is an affordable solution for IoT systems built from such devices.

We showed that T-RAID has similar security properties to those of HYDRA, a secure remote attestation scheme proposed in the past. T-RAID, however, performs more complex integrity checks on the device aiming at detecting rootkits both in memory and in persistent storage. While our integrity checks effectively detect malware, unfortunately, our current prototype implementation of T-RAID has some weaknesses stemming from limitations of OP-TEE, the TEE implementation that we use, and the inherent parallelism provided by multi-core processor architectures.

References

1. Alrawi, O., Lever, C., Antonakakis, M., Monrose, F.: SoK: security evaluation of home-based IoT deployments. In: IEEE Symposium on Security and Privacy, pp. 1362–1380 (2019). https://doi.org/10.1109/SP.2019.00013
2. Antonakakis, M., et al.: Understanding the Mirai botnet. In: USENIX Security Symposium, pp. 1093–1110. USENIX Association, August 2017
3. Castelluccia, C., Francillon, A., Perito, D., Soriente, C.: On the difficulty of software-based attestation of embedded devices. In: ACM Conference on Computer and Communications Security (CCS), pp. 400–409 (2009). https://doi.org/10.1145/1653662.1653711
4. Cozzi, E., Vervier, P.A., Dell'Amico, M., Shen, Y., Bigle, L., Balzarotti, D.: The tangled genealogy of IoT malware. In: Annual Computer Security Applications Conference (ACSAC) (2020)

5. Eldefrawy, K., Rattanavipanon, N., Tsudik, G.: HYDRA: hybrid design for remote attestation (using a formally verified microkernel). In: ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec), pp. 99–110 (2017). https://doi.org/10.1145/3098243.3098261
6. Eldefrawy, K., Tsudik, G., Francillon, A., Perito, D.: SMART: Secure and minimal architecture for (establishing a dynamic) root of trust. In: Network and Distributed Systems Symposium (NDSS) (2012)
7. ENISA: Guidelines for securing the Internet of Things. ENISA study, November 2020
8. ETSI: CYBER; Cyber security for consumer Internet of Things: Baseline requirements. ETSI TS 103 645 v2.1.2, June 2020
9. Francillon, A., Nguyen, Q., Rasmussen, K.B., Tsudik, G.: A minimalist approach to remote attestation. In: Conference on Design, Automation & Test in Europe (DATE), pp. 1–6 (2014)
10. Global Platform: Security evaluation standard for IoT platforms v1.1 (SESIP). Global Platform Standard, June 2021
11. Gu, J., Xian, M., Chen, T., Du, R.: A Linux rootkit improvement based on inline hook. In: Proceedings of the 2nd International Conference on Advances in Mechanical Engineering and Industrial Informatics, pp. 793–798. Atlantis Press (2016). https://doi.org/10.2991/ameii-16.2016.155
12. Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., Sikdar, B.: A survey on IoT security: application areas, security threats, and solution architectures. IEEE Access **7**, 82721–82743 (2019). https://doi.org/10.1109/ACCESS.2019.2924045
13. Kumar Jain, V., Gajrani, J.: IoT security: a survey of issues, attacks and defences. In: Sharma, H., Saraswat, M., Kumar, S., Bansal, J.C. (eds.) CIS 2020. LNDECT, vol. 61, pp. 219–236. Springer, Singapore (2021). https://doi.org/10.1007/978-981-33-4582-9_18
14. Li, Y., Cheng, Y., Gligor, V., Perrig, A.: Establishing software-only root of trust on embedded systems: facts and fiction. In: International Workshop on Security Protocols, pp. 50–68 (2015). https://doi.org/10.1007/978-3-319-26096-9_7
15. Nagy, R., Németh, K., Papp, D., Buttyán, L.: Rootkit detection on embedded IoT devices. Acta Cybernetica, August 2021. https://doi.org/10.14232/actacyb.288834
16. NIST: Considerations for managing Internet of Things (IoT) cybersecurity and privacy risks. NISTIR 8228, June 2019
17. NIST: Baseline security criteria for consumer IoT devices. NIST draft white paper, August 2021
18. Seshadri, A., Perrig, A., van Doorn, L., Khosla, P.: SWATT: software-based attestation for embedded devices. In: IEEE Symposium on Security and Privacy, pp. 272–282 (2004). https://doi.org/10.1109/SECPRI.2004.1301329
19. Seshadri, A., Luk, M., Shi, E., Perrig, A., van Doorn, L., Khosla, P.: Pioneer: verifying code integrity and enforcing untampered code execution on legacy systems. ACM SIGOPS Oper. Syst. Rev. **35**(5), 1–16 (2005). https://doi.org/10.1145/1095809.1095812
20. Tamás, C., Papp, D., Buttyán, L.: SIMBIoTA: similarity-based malware detection on IoT devices. In: Proceedings of the 6th International Conference on Internet of Things, Big Data and Security - IoTBDS, pp. 58–69. SciTePress (2021). https://doi.org/10.5220/0010441500580069

Secure Authentication for Everyone! Enabling 2nd-Factor Authentication Under Real-World Constraints

Julian Fietkau[1](✉), Syeda Mehak Zahra[1], and Markus Hartung[2]

[1] Technical University of Berlin, Berlin, Germany
fietkau@tu-berlin.de
[2] Avira Operations GmbH, Tettnang, Germany
markus.hartung@avira.com

Abstract. Millions of user accounts have been exposed by data breaches within the last years. The leaked credentials pose a huge threat to many because they can be used for credential stuffing and brute-force attacks across all online services. The best solution for this problem seems to be the use of 2nd-factor authentication, like hardware tokens or one-time passwords. While these are great solutions, they cause many problems for users because they are too expensive, difficult to manage, or just not user-friendly. In this paper, we will present the results of a study that shows that users need and want secure authentication, as long as it is quick, easy, and free of charge. Hence, we investigate how recent advancements in smartphone security and authentications standards can be used to build a mobile authenticator that is easy to use, free of charge, and as secure as a hardware token. Therefore we leverage the Trusted Execution Environment of the Android platform to implement a FIDO compliant authentication mechanism on the smartphone. Furthermore, we integrate this mobile authenticator into a password manager app, to reduce user interaction, simplify the setup and provide an encompassing solution for the user.

Keywords: 2nd-factor authentication · Data breaches · Leaked credentials · Fast IDentity Online · Trusted execution environments · Secure logins · Low cost security · Password manager · Biometric authentication

1 Introduction

The number of leaked credentials caused by data breach attacks has been increased tremendously over the past years [16,20,23]. According to our statistics, there is a steady increase in the number of breaches since 2005 and an increase of 45% just in the year 2017 [15]. Moreover, many users use the same or a similar password for every service, hence more than just the targeted service is under threat [16]. To tackle this issue, companies and researchers are trying their best to secure user logins against password stuffing attacks.

E. Gelenbe et al. (Eds.): EuroCybersec 2021, CCIS 1596, pp. 89–101, 2022.
https://doi.org/10.1007/978-3-031-09357-9_8

One of the most promising solutions to this problem is 2nd-factor authentication. Here, Users have to prove their identity twice during the login process by providing their password and a 2nd-factor such as hardware tokens, mobile TANs, or providing a cryptographic signature. The FIDO Alliance and its biggest partners, e.g. Google, are playing a key role in this fight to make authentication more secure, e.g., by promoting the use of hardware tokens [11]. Hardware tokens are small little devices, just like USB thumb drives, that can be connected to most devices to enable 2nd-factor authentication. While the cryptographic fundament is very solid and the overall idea of hardware tokens is outstanding, we suppose that hardware tokens don't scale. The reason for this assumption becomes clear, even before a user is ready to use it. The average price of a security token is somewhere between \$30–\$70 [25]. This price tag appears small for some people but becomes problematic if we imagine worldwide adoption. It becomes even worse when we incorporate that a single token is not enough, since users need to have backup tokens in case they lose or break these devices. As we see the threat of leaked credentials is growing and 2nd-factor authentication is providing a good solution for some, but we have to admit that solutions like hardware tokens might not scale for everyone due to their costs and management overhead.

Hence, in this work we want to show how a mobile authenticator can be built without the costs of an additional hardware device, but with similar security standards. Therefore we combine recent advancements of FIDO standards and new security features of the Android mobile operating system, to build a mobile authenticator that can be used on every FIDO compliant web service. We will explain how to implement such a mobile authenticator, how the underlying technology works and why we consider it secure. In summary, we make the following contributions:

- We discuss the issue of data breaches and how they are threatening all online services, not only those that have been attacked recently.
- We compare the current approaches for 2nd-factor authentication, discuss their limitations and identify the main reasons for the lack of adoption.
- We design and implement a mobile authenticator that combines the most recent advancements of the FIDO authentication standard and Android Security, to enable secure authentication for everyone.
- To simplify the overall process, we integrate our solution into the Avira Password Manager, which is freely available via the following link: https://www.avira.com/en/password-manager

2 Background

This section provides the necessary background knowledge about password security, data breaches, authentication, and security keys.

2.1 The Data Breach Problem

A data breach is a disclosure of private and confidential information. During the past few years, these incidents have increased tremendously. According to 'Have I Been Pwned', 11 billion user accounts from roughly 550 different websites have been compromised by data breaches so far [23]. The rising number of leaks is the result of the surging number of hacking attempts - automated phishing, malware, and brute-force attacks somewhere hit a target and allow the attacker to gain unauthorized access to databases full of user credentials [16]. When this data becomes public, it becomes a big issue for all online services, since many people reuse the same credentials on multiple services or use password patterns that are simple to guess [6]. This is how data breaches become a threat for countries, individuals, and big organizations. For example, affected companies may have to compensate their customers, become incapable of acting for weeks or months, and can even face court. Worst of all, the loss is unpredictable and can be low or high. One study from 2018 estimates that the average cost of a data breach in the U.S. is around $7.91 million, and almost 30% of all companies lose revenue after a data breach [18]. As we see, data breaches can have a huge impact, not only in a financial way but also on the operation, reputation, and image of an organization. A well-known solution to tackle this problem is to secure the logins with strong 2nd-factor authentication.

2.2 Hardware Authenticators

One way to integrate 2nd-factor authentication are Hardware Authenticators. By adding a 2nd layer of protection, an attacker can not log into a leaked account without the Hardware Authenticator device and the secret key it contains [19]. Hardware Authenticators are easy to use because they don't require batteries or some kind of additional software in order to run nowadays. On the other hand, a stolen or lost authenticator is an organizational disaster and can only be mitigated by adding multiple authenticators [19]. Several companies are producing Hardware Authenticators like Yubico, Kensington, and Thetis [1,25]. They are available in different price ranges, starting from $30 upwards. Security-Tokens are largely adopted by some organizations, e.g., Google, Facebook, etc., and have proven to be useful in practice [24]. The main reason for the great success of these tokens is the cooperative work of the FIDO Alliance that defines and maintains this open and independent technology for everyone.

2.3 FIDO

The Fast IDentity Online (FIDO) Alliance came into existence to promote new authentication standards and reduce the use of passwords [21]. Because this is an issue of many, the open industry association is supported by big companies, e.g., Amazon, Facebook, Google, Microsoft, and American Express [9]. FIDO covers a large number of technologies, including security tokens, smart cards, NFC, communication standards, and also biometrics such as fingerprint, iris,

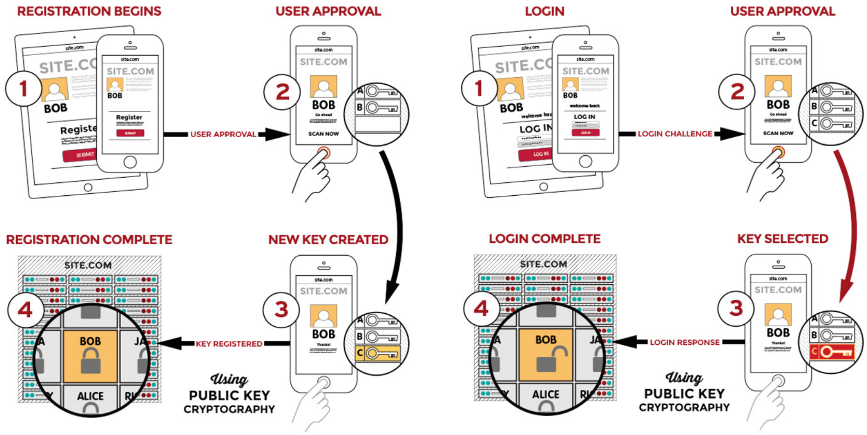


Fig. 1. Registration and authentication process as defined by FIDO [11]

voice, and facial recognition. The core of FIDO mainly establishes the following two processes:

i) Registration: A user receives a unique username and a randomly generated challenge. Depending on that, the user authenticator can generate a public and private key. This public key and some metadata are stored by the service [21].
ii) Authentication: The user sends the given username to the service and receives a new challenge. This challenge will be signed by the authenticator using its private key and sent back to the service afterward. The service can validate this signature using the public key of the user to verify its identity [21].

The most relevant parts of the FIDO specification for this work are FIDO UAF, U2F, and CTAP [11]. FIDO Universal 2nd-factor (U2F) specifies a universal 2nd-factor experience. The Universal Authentication Framework (UAF) defines the use of native device features like biometric authentication, e.g., fingerprint or face recognition [12]. FIDOs Client To Authenticator Protocol (CTAP) describes how the OS and a browser, can establish a connection with external devices via Bluetooth (BLE), Near Field Communication (NFC) or USB [10].

3 How to Solve the Data Breach Problem?

Several studies show, that breaches are getting more extensive and more frequent [6,15]. In 2019 alone, there have been at least four major data breaches, each impacting more than 200 million records. One of these, known as Collection #1, contains more than 2.7 billion email password pairs [6] and is one of the largest data breaches on the Internet. Experts have reviewed the collection and concluded that the list combined 2000 previous data breaches and added

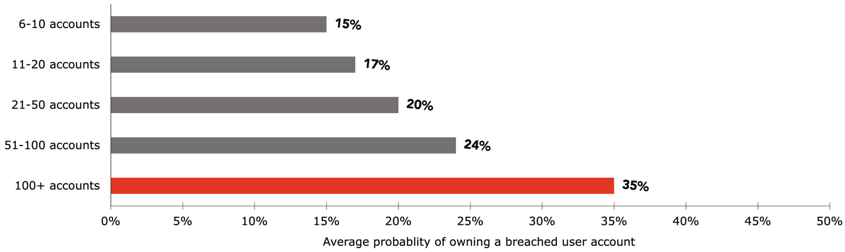


Fig. 2. More accounts, more breaches: The probability of owning a account that is part of a data breach increases with the number of accounts the user owns [7].

an estimate of 140 million new email addresses and 10 million new passwords from unknown sources [8]. In fact, for many breaches, it's not clear where data originates, because data get hacked, scrapped, and dropped in so many ways. Sometimes hackers are selling label the data with the name of the affected company, sometimes the data is assembled from various data breaches, and in other cases, the data is dropped without any further means. It's also not clear, why all the data get hacked in the first place because companies are often not able to detect the breaches and avoid speaking about it. But we know, there are a lot of bad security habits, such as weak and recycled passwords across various accounts, badly maintained software that can be exploited, and poorly secured databases. For example, leaked passwords are often available in plaintext rather than in their hashed version [23]. One reason might be, that passwords are not handled properly in the first place (hashing and salting passwords before they get stored in a database). Another reason is, that hackers might be able to decrypt them because weak or broken hashing algorithms have been used.

Whatever the reasons are, we often don't know, but we can measure its impact. As shown in Fig. 2, the more accounts a person owns, the higher the probability that they will be hacked. This is the result of a study conducted by Avira in early 2019 [6]. The data tell us, that users with 6 to 10 accounts have a 15% chance of a breach. This probability jumps up to 35% when the number of accounts is 100+. The main reason behind this increase seems to be the heavy reuse of account names and passwords across various online services. To follow up on this, Avira conducted an online survey with 2519 respondents aged between 20–65 years in the US [6]. A key insight from this study is, that users are more interested in simplifying authentication rather than just securing it. When they have been asked for reasons to adopt password managers, 48% of the participants said that they would adopt password managers, if they can log in more quickly and easily. A few less (44% of the participants) have indicated that they would use it to protect their passwords against hackers. Nevertheless, another study from 2019 by Pearman et al., tells us that most people don't want to pay for a password manager solution and prefer to use a free version. Just a few people said they might be willing to pay for this, and only if the tool was very secure and very easy to use [22].

In summary, science is telling us that we need to adopt more secure authentication, since data breaches and leaked accounts are threatening all online services and their users. While people seem to know about the issue, they are only ready to adapt if things are easier to use and will be free of charge.

4 A Secure Mobile Authenticator for Everyone

In this section, we explain how our mobile authenticator works and how we connected the various advancement of FIDO and Android OS Security to build a no-cost mobile authenticator for everyone.

Considering the recent advancements in phone security, a modern smartphone is in many ways as secure as a hardware authenticator. It can securely store private keys within the secure key storage [3] and the Trusted Execution Environment represents dedicated hardware, with well-defined cryptographic algorithms, offering just a limited attack surface [5]. In other ways, a smartphone is even superior to hardware tokens: A phone can be updated, is always with you and people already know how it's used. A hardware token on the other hand requires some effort to know how it works, you can lose or break it and updates are not supported or rather complicated. Moreover, many smartphones offer sophisticated algorithms for local authentication, e.g., finger or face recognition, something a cheap hardware token can not offer with the same level of security and useability. Finally, another great feature is to remotely find, lock, or erase the phone in case of losing it or when it gets stolen [13]. To the best of our knowledge, no hardware token offer such features, hence a stolen token requires a user to invalidate the keys and manually regain access to his accounts. Hence, we want to combine the most recent advancements in authentication standards and smartphone security to create a mobile authenticator that enables secure authentication for everyone. While keeping the same security level, as given with hardware tokens, we remove the main drawbacks such as ease of use, updatability, advanced local authentication, remote deletion, and most of all the additional costs, which are major reasons for the lack of adoption.

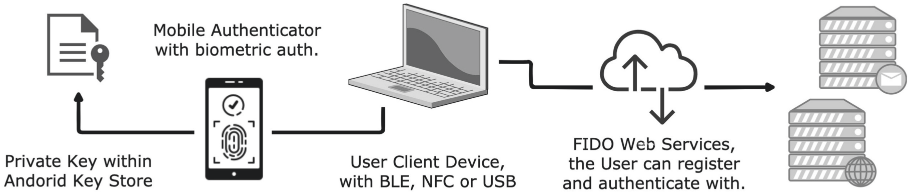


Fig. 3. Mobile authenticator

In Fig. 3, we show how we imagine a mobile authentication setup. While the underlying technology is much more complicated, we want to initially focus on the abstract view on the applications layer with a user's perspective in mind.

The setup comprises certain devices, owned and managed by two parties: a user owning a client device and a smartphone with the mobile authenticator installed. On the other side, a server, owned by the service that supports FIDO compliant authentication. This service can be anything from cloud services, social networks, or just a simple mail service. The owner of the service needs to implement FIDO compliant authentication. Therefore, he can use several FIDO certified third-party products, e.g. WebAuthn Awesome, that need to be integrated into the provided service [2]. While it requires some effort, it's probably one of the best ways to protect the service and its customers.

The communication between the components, shown in Fig. 3, can be established as follows. The client device has a secure HTTPS connection to communicate with the service he wants to authenticate with. In addition, another secure connection via BLE, NFC, or USB is created connection with the smartphone. The smartphone application, storing the private keys within the secure key storage, can not directly access the keys, instead, it needs to authenticate and communicate with the Trusted Execution Environment to execute cryptographic operations that use the key. When the components have been assembled in the right way, the registration and authentication procedures are ready to go.

Both registration and authentication, require 4 steps, as shown in Fig. 1. To implement these operations, we build four Java modules i) a module to establish a Bluetooth connection; ii) a module to encode, send and receive FIDO messages; iii) a module to use the cryptographic operations of the secure TEE; iv) a module to protect the access of the authenticator using biometric authentication. Using these modules we have implemented two generic functions that can perform FIDO registration and authentication procedures. Some of the implementation details can be described by stepping through the typical use cases. Please note, that we discuss the implementation for the smartphone only because the user device and web service are not part of our work.

- **Device connection:** A secure connection using BLE is created between the user device and the smartphone hosting the authenticator. For secure pairing of both devices, we had to implement an android BLE class using a GATT server. Once the devices have discovered each other, the client device gets some connection information from the authenticator and can pair the devices. When starting a 2nd-factor authentication, the browser will automatically search for a connected authenticator device, e.g., via NFC, BLE, USB.
- **Local Authentication:** The mobile authenticator app needs to be installed on the smartphone. The phone owner needs to authenticate every time he wants to use the app. The local authentication has been implemented with the face authentication procedures of the BiometricPrompt API. It supports authentication using the user's finger or iris, depending on which property is enrolled by the user.
- **Secure Key Storage:** A user can register the authenticator with any FIDO compliant service, after a successful login or even during account creation. During the registration, the authenticator needs to be unlocked and consent must be given to generate a new key pair. To generate the cryptographic keys

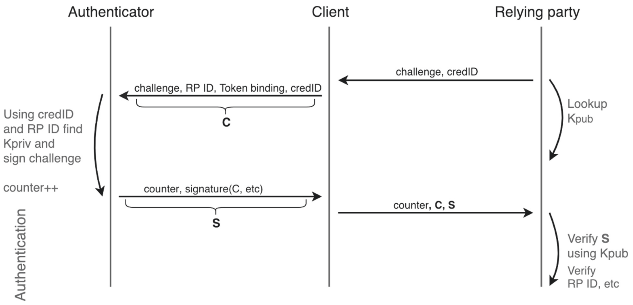


Fig. 4. Message flow between authenticator, client device and the service [10]

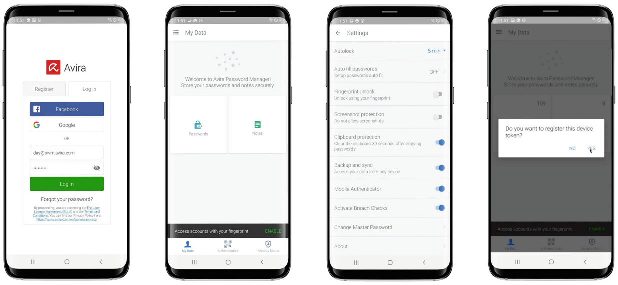
we have used the ECPublicKey class to generate a public and private key and to directly stored the private key within the Android Keystore. During registration, the public key is transferred to the service using the message API. Every registration will start another key setup and repeat the previously described steps.

- **Authentication:** After registration, the user will be asked to provide the 2nd-factor on every new login attempt. At this point, the authenticator needs to be unlocked using the local authentication feature. The authenticator sends the username and implicitly requests a challenge from the service with the message API. The message flow of this procedure is shown in Fig. 4. After receiving the challenge, the authenticator needs to look up the private key in the key manager of the Android Keystore. To sign the message within the Android Keystore we used the KeyStore signMessage() method. Afterward, the signature is passed to the app and the message API is used to transfer the signature and some additional metadata to the service. A final response will indicate if the access is granted or not.

4.1 Integration of the Authenticator into a Password Manager

Another idea we want to present is to integrate the mobile authenticator into a password manager application (PWM). A PWM is a tool that can create, store, and enter passwords for you in a secure way. It will store the passwords within a cryptographically secured file, that can only be accessed by entering a master password to protect the data from unauthorized access. Nowadays, not only password but all kinds of data can be stored, such as credit card information, notes, images, etc. Most vendors further provide browser extensions, smartphone apps, and online backup features to make it very convenient to use. Some popular examples are Enpass, Avira Password Manager, Bitwarden, Authy, LastPass, and 1Password.

We think, that the integration of the mobile authenticator into a PWM makes sense for two reasons: First, the user might already use the PWM to access the credentials for the first authentication step. Second, we can reduce the number



Welcome to Avira Password Manager!
Store your passwords and notes securely.

Access accounts with your fingerprint ENABLE

By proceeding, you are accepting the End User License Agreement (EULA) and the Terms and Conditions. You can find our Privacy Policy here: https://www.avira.com/en/general-privacy.

das@pwm.avira.com

Settings

Autolock 5 min

Auto fill passwords — Setup passwords auto fill — OFF

Fingerprint unlock — Unlock using your fingerprint

Screenshot protection — Do not allow screenshots

Clipboard protection — Clear the clipboard 30 seconds after copying passwords

Backup and sync — Access your data from any device

Mobile Authenticator

Activate Breach Checks

Change Master Password

About

Do you want to register this device token?

Fig. 5. Integration of the mobile authenticator into the avira password manager

of apps used which improves user confidence and makes it more quick and easy to use. Only a single app needs to be installed and managed to grant access to your credentials and to unlock the 2nd-factor capabilities. Furthermore, only a single authentication, using finger or face recognition, is required to unlock all the secrets and access the mobile authentication functionality.

For these reasons, we integrated the mobile authenticator into the Avira Password Manager, as shown in Fig. 5. The mobile authenticator can be enabled within the settings of the PWM. Afterward, the authenticator is active in the background and will communicate with the user via different prompts.

To implement the mobile authenticator we used Java and Kotlin, the standard programming language for Android development. Furthermore, we have used the following libraries to build our solution. Jackson data-bind library is used for data binding, which is used to convert JSON to and from plain java objects. For the UI integration of the Authenticator, we used the Android Material Design library, which provides some easy-to-use front-end widgets. Google guava API providing an advanced Java collection framework and offers a lot of handy features for functional programming, range objects, and hashing.

5 Discussion

In this section, we want to have a short discussion on the security of the implemented authenticator. Furthermore, we want to discuss how the authenticator compares to one-time passwords (OTP), which are often used to implement 2nd-factor authentication while avoiding the various drawbacks of hardware tokens.

5.1 Authenticator Security

Considering the application of our mobile Authenticator, we have to discuss its security. Our solution mainly relies on the following three security features of the Android platform security: secure key storage, strong and secure cryptographic algorithms, and a secure generation of cryptographic keys. The Android Keystore system lets you store cryptographic keys in an isolated subcomponent, called the Trusted Execution Environment (TEE), to make it very difficult to extract key material from the device [3]. The key material is never exposed outside the TEE. If the Android OS is compromised, e.g., an attacker can read the device's memory, the attacker may be able to use any keys on the device, but can not extract it from the device. Hence, once the keys are in the Keystore, they are secure and can only be used with dedicated cryptographic operations. In addition, this operation is restricted to authenticated users only, which requires local authentication of the device owner. The Hardware security module contains only well-known and largely tested cryptographic algorithms, that are considered secure and state-of-the-art [4]. It also provides a dedicated true random-number generator to generate cryptographic keys with sufficient entropy. Furthermore, mechanisms such as resist package tampering and countermeasures against unauthorized side-loading of apps are in place to mitigate various memory attacks [5]. In summary, the Android OS includes very advanced features to provide a high level of security to protect the user's data and the mobile authenticator. Since the key will never leave the TEE, a lot of security measures are in place to prevent the key extraction. Outside of this secure environment, the communication between the devices will be secured with secure BLE pairing and HTTPS. Beyond that, when a data breach will affect one of the registered services, it can not leak any new user credentials, because only a public key and a random username is stored there.

5.2 Comparison with OTP

One-time passwords are 2nd-factor solutions, that are based on a shared secret and a hash function to generate new and unique passwords [17]. Comparable to the mobile authenticator, this solution requires registering a dedicated hardware or software solution with the service that can securely store the shared secret. Using this shared secret a derived password can be calculated, which can be used to authenticate to the service [17]. In the past, a lot of companies have implemented OTP, to add a 2nd-factor without the downsides of hardware tokens.

When comparing both solutions, OPT and the authenticator, are very similar but each of them has some advantages and disadvantages. Both systems rely on the availability of the device. While OTP does work even without the Internet, the authenticator requires Internet and a local connection via BLE, NFC, or USB. On the other hand, this makes the authenticator easier to use, because OTP typically requires to enter the 2nd-factor by hand and does not exchange the authentication information automatically [17]. While OTP can be synchronized with various apps like Google Authenticator, etc., the implemented authenticator is device-specific and requires a Trusted Execution Environment and advanced biometric authentication features.

Both solutions are very secure, but we think there are some major drawbacks for OTP. An attacker breaching a large authentication database will be able to generate valid OTP values at his will. With our authenticator, only a public key will be leaked, which does not threaten the user at all. On the other hand, a lost or broken OTP token can be easily replaced with just another one that only needs to be synchronized with a service [14]. A lost authenticator instead, will be a real disaster. Once the authenticator is lost or broken, the keys can never be extracted or recovered. When changing the smartphone device, it is necessary to deregister or deactivate the authenticator for each service on its own. A centralized deregistration of all authenticator tokens managed by the app could be implemented within a PWM solution. This could help a user to disable all 2nd-factor tokens managed by the current phone before deactivating or reselling it. When activating a new phone, the user can start again to register a new authenticator with the services, without the issue of being logged out.

6 Conclusion

In the following, we will summarize our work and discuss the milestones we have achieved. The goal of our work was to investigate how the most recent advancements in FIDO specifications and smartphone security can be leveraged to build a secure mobile authenticator on the smartphone. To motivate our work, we discussed the various issues related to data breaches and presented some insights and statistics that have been collected from Avira's password manager. The data shows clearly that 2nd-factor authentication is a strong requirement nowadays. Hardware tokens are one way to implement this and they can prevent credential stuffing and brute-force attacks that can be affiliated to the rising number of leaked credentials and data beaches [23]. While hardware tokens are a great solution for many, they face some major issues when talking about worldwide adoption. Hence, we build a mobile authenticator that connects the most recent advancements in authentication standards and smartphone security, to enable secure 2nd-factor authentication without additional hardware cost. We discussed how to implement such a solution within the Android Trusted Execution Environment and how to integrate it within the Avira Password Manager to make the user experience more seamless and reduce user interaction. For the evaluation, we have reviewed our solution in terms of security and we compared it with one-time passwords, which are often used to implement 2nd-factor authentication without additional hardware costs. Based on our work, we perceive the mobile authenticator to be a robust, secure, and easy-to-use replacement for hardware authenticators, which can reduce the key disadvantages of hardware tokens, namely costs, management overhead, and usability.

Acknowledgements. The authors want to thank the Review Committee for the valuable feedback and comments. The project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 952684. Opinions, views, and conclusions are those of the authors and do not reflect the views of anyone else.

References

1. Etienne, S.: The best hardware security keys for two-factor authentication (2019). https://www.theverge.com/2019/2/22/18235173/

2. Yuriy, A., various GitHub contributors.: Webauthn awesome. a repository with awesome webauthn/fido2 resources (2020). https://github.com/herrjemand/awesome-webauthn
3. Android Open Source Project: Android keystore system (2020). https://developer.android.com/training/articles/keystore
4. Android Open Source Project: Trusty tee (2020). https://developer.android.com/guide/topics/security/cryptography
5. Android Open Source Project: Trusty tee. androids trusted execution environment (2020). https://source.android.com/security/trusty
6. Avira Operations GmbH & Co. KG: Avira Password Security Report. Tidy up your digital life. (2019). https://www.avira.com/files/press/2019/PasswordSecurityReport_EN.pdf
7. Avira Operations GmbH & Co. KG: Avira Privacy Report. Tidy up your digital life. (2019). https://www.avira.com/files/press/2019/AviraSummerPrivacyReport_Final.pdf
8. Brian Barrett for wired.com: Hack brief: An astonishing 773 million records exposed in monster breach (2020). https://www.wired.com/story/collection-one-breach-email-accounts-passwords/
9. Octopus, C.: Fido 101: Understanding fido strong authentication and what it can do for you (2018). https://blog.strongkey.com/blog/fido-101-strong-authentication
10. Fast IDentity Online (FIDO) Alliance: Fido client to authenticator protocol (CTAP) specification (2018). https://fidoalliance.org/specs/fido-v2.0-id-20180227/fido-client-to-authenticator-protocol-v2.0-id-20180227.html
11. Fast IDentity Online (FIDO) Alliance: How fido works (2020). https://fidoalliance.org/how-fido-works/
12. Fast IDentity Online (FIDO) Alliance: Specifications overview (2020). https://fidoalliance.org/specifications/
13. Google: Find, lock, or erase a lost android device (2020). https://support.google.com/accounts/answer/6160491?hl=en
14. Grimes, R.A.: Hacking Multifactor Authentication. Wiley, Hoboken (2020)
15. Johnson, J.: Annual number of data breaches and exposed records in the US from 2005 to 2020 (2021). https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches
16. De Groot, J.: The history of data breaches (2019). https://digitalguardian.com/blog/history-data-breaches
17. Liu, Z.: An improved one-time password authentication scheme. In: 2013 15th IEEE International Conference on Communication Technology, pp. 1–5 (2013)
18. Davis, M.: 4 damaging after-effects of a data breach (2019). https://www.cybintsolutions.com/4-damaging-after-effects-of-a-data-breach/
19. Tillman, M.: What are security keys, how do they work, and which is the best to buy? (2020). https://www.pocket-lint.com/gadgets/news/150395-best-hardware-security-keys-for-two-factor-authentication
20. Mozilla Foundation : Archive of all breaches in Firefox Monitor (2020). https://monitor.firefox.com/breaches
21. Newhouse, W., Johnson, B., Kinling, S., Kuruvilla, J., Mulugeta, B., Sandlin, K.: Multifactor authentication for e-commerce: Risk-based, fido universal second factor implementations for purchasers. Technical Report, National Institute of Standards and Technology (2019)
22. Pearman, S., Zhang, S.A., Bauer, L., Christin, N., Cranor, L.F.: Why people (don't) use password managers effectively. In: Fifteenth Symposium on Usable Privacy and Security ({SOUPS} 2019), pp. 319–338 (2019)

23. Hunt, T.: Have I been pwned? (2018). https://haveibeenpwned.com
24. Yubico: Google defends against account takeovers and reduces it costs (2018). https://resources.yubico.com/53ZDUYE6/as/q3unyy-dmr8u0-fds0yi/Google_Case_Study.pdf
25. Yubico: Yubikey 5. for businesses, professionals and individuals (2020). https://www.yubico.com/de/store/#for-professionals

Energy, QoS and Security Aware Edge Services

Erol Gelenbe[1,2](✉), Mateusz P. Nowak[1], Piotr Frohlich[1], Jerzy Fiolka[3], and Jacek Checinski[3]

[1] Institute of Theoretical and Applied Informatics, Polish Academy of Sciences,
Baltycka 5, 44-100 Gliwice, Poland
gelenbe.erol@orange.fr

[2] LabI3S, Universite Cote d'Azur, Grand Chateau, 06103 Nice, France

[3] Faculty of Automatic Control, Electronics and Computer Science, The Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland
http://www.iitis.pl

Abstract. With the development of communication technologies and the increasing bandwidth of optical fibres and transmission speeds in current 5G and future 6G wireless networks, there is a growing demand for solutions organising traffic in such networks, taking into account both end-to-end transmissions and the possibility of data processing by edge services. The most pressing problems of today's computer networks are not only bandwidth and transmission delays, but also security and energy consumption, which is becoming increasingly important in today's climate. This paper presents a solution based on neural networks that organises network traffic taking into account the above criteria - quality of service (QoS), energy consumption and security.

Keywords: SDN · Random Neural Networks · Green computing · Edge computing · Energy-awareness · Green networking · Security · IoT · QoS

1 Introduction

Today's communication technologies are capable of transmitting increasing amounts of data per second. Their source is not only the data of human-operated applications, but increasingly the sensors and hubs of major applications such as healthcare [6,31] and the of the Internet of Things (IoT) and other services. However, the Internet's ease of use and high bandwidth also creates tremendous opportunities for attackers, so that all these Internet accessible systems need to be protected from malicious attacks [5,32].

Since the computational capabilities of servers and workstations are limited and they are not always able to process data at an appropriate speed, Cloud

This work has been supported by the European Commission H2020 Program through the IoTAC Research and Innovation Action, under Grant Agreement No. 952684.

E. Gelenbe et al. (Eds.): EuroCybersec 2021, CCIS 1596, pp. 102–117, 2022.
https://doi.org/10.1007/978-3-031-09357-9_9

architectures have become the answer to this problem, grouping servers into structures that provide huge computing capacities, but these need to be properly accessed and scheduled [4,40,42]. The second trend, which is gaining momentum especially with the development of 5G networks, is the multiplication of computing services and their movement to the Edge, close to the users and to the sources of data.

The primary purpose of a computer system aand network is to process and transmit data while maintaining adequate Quality of Service (QoS) [20]. Disturbances in QoS result in the need to wait for data, thus wasting computing power, and often in the need to resend data, which in addition, in the case of IoT devices, is associated with energy expenditure and shortening the life of a battery-powered device. QoS problems could be avoided if it were possible to place processing nodes close enough to the data source so that transmission would not be a problem. However, this can be too costly, both at the investment stage and later when it comes to covering energy costs. Electricity, apart from being an obvious cost for the operator, is obtained in the overwhelming majority from non-renewable energy sources, and its unnecessary consumption has an impact on the climate of our planet. It should therefore be saved for both economic and ecological reasons [22]. How important, although underestimated, is 'green computing' and 'green networking' [3,35] is shown by the fact that, at present, the energy consumption of IT systems accounts for roughly 10% of global electricity consumption, and by 2030 this share may even reach 20% [1,16].

Another problem is security which needs to be assured [19,21]. As the value of data transmitted over the network and processed on external servers increases, so do the number of attacks on the infrastructure for transmitting, processing and storing information. Modern computer systems must take this issue into account already at the design level, according to the security-by-design principle.

Our ppaer addresses all these issues, improving network performance in terms of QoS, power consumption and security [27]. This article is composed of six main sections. In Sect. 2 we briefly introduce the reader to the topic of RNNs, referring to previous publications on the subject. We show the specifics of the environment that is the subject of the current research and the tailored solutions that we have used. Section ?? discusses how to collect QoS, energy and security data that RNNs use to make decisions. Section 3 presents the experimental part, including a description of the implementation and the testbed. It also includes a discussion of the obtained results. The whole work is summarised in the Sect. 4.

2 Random Neural Networks for the Control of Computer Networks

The optimization the QoS of distributed systems has been discussed in numerous publications [28,38,39,41,42]. QoS versus energy consumption of distributed services has also been examined experimentallly in [18]. However, the focus on security is more recent and its impact on network management and routing is examined in [10,11,17].

To control the network in terms of multiple criteria, including QoS, security and energy in our case, we use a solution based on Random Neural Networks (RNNs) [12,13], trained using Reinforcement Learning. RNNs optimize data packet transmission paths as well as the selection of Egde Computing services in such a way as to maintain an appropriate (predefined) balance between QoS, energy consumption and security. The switches and servers of a computer network form a distributed system, and its optimization is a variation of a well-known problem. However, by using the RNN and placing our system in a Software Defined Network (SDN) environment as in [8,9], we show that familiar Machine Learning techniques can also be used in state-of-the-art network architectures.

It should be noted, however, that the use of an SDN controller to implement the presented solutions is convenient from the point of view of demonstrating the usefulness of RNN in computer network control, but due to the distributed architecture of the RNN-based Decision Engine the same solutions can - under certain conditions - also be applied to a traditional, fully distributed network architecture.

The problems of SDN design and optimization are discussed in survey paper [36], taking into account not only energy efficiency issues, but also touching on security problems. Security issues in SDN have received a number of publications, for example in [2]. An interesting survey article on system deployment and optimization, shedding light on our work, was published in [25]. The popularity of this technology and the ease of implementation of routing control algorithms are also significant.

2.1 The Goal of the Decision System

The system we consider consists of:

- The set of network SDN switches or forwarders $S = \{s_1, \;..\; s_n\}$ that are interconnected via a network graph, where S is the set of nodes and A is the $n \times n$ one-hop binary connection matrix between nodes.
- Every switch $s \in S$ may have connected "clients" or Edge services.
- The set of Clients is $C = \{c_1, \;...\; c_m\}$ and each client c has a node or switch $s(c)$ to which it is directly connected .
- Edge services are used to offload specific cloud services (with their processing capacity and/or repositories) that are operating in close proximity so as to offer fast service to the clients. They belong to a set $E = \{e_1, \;...\; e_M\}$ of M services which all offer equivalent facilities in terms of processing and the ability to provide specific data. Also any service e is connected to some switch or node $s(e)$.

The Goal of the decision system is to find a P among the set of switches S to connect the pair of clients $(c, c'),\; c, c' \in C$ or the client-service pair $(c, e),\; c \in$

$C,\ e \in E$. The choice of the path is based on the QoS, security and energy criteria, or one or two of these criteria. For ease of notation we will denote a connection (c, c') or $(, e)$ as a "flow" f.

Thus a path:

- $P = P(c, c')$ from c to c' is $P(c, c') = (s(c), s(P)_1,\ \dots\ .s(P)_{l(P)-2}, s(c'))$, or
- A path $P = P(c, e)$ from c to e is $P(c, e) = (s(c), s(P)_1,\ \dots\ .s(P)_{l(P)-2}, s(e))$, where
- $A(s(c), s(P)_1) = 1$, $A(s(P)_i, s(P)_{i+1}) = 1$, for $1 \leq i \leq l(P) - 3$, $A(P_{l(P)-2}, s(c')) = 1$, $A(P_{l(P)-2}, s(e)) = 1$,
- and $l(P)$ denotes the length of the path P in number of switches or nodes.

Thus we can now formulate the goal function G for given flow and path as the weighted sum of three criteria:

$$G(f, P) = aQ(f, P) + bT(f, P) + cJ(f, P), \tag{1}$$

where $a,\ b,\ c$ are non-negative constants with $a + b + c = 1$, and $Q(f, P)$ is the QoS value for given flow f using path P. For instance, $Q(f, P)$ can be the end-to-end delay per packet for flow f on path P or the corresponding packet loss, or some combination thereof. The measurement of such metrics is presented in Section ?? below.

$T(f, P)$ is the trust metric that expresses the level of insecurity of traffic belonging to given flow f going along the path P. It can be obtained via Attack or Anomaly Detectors, Honeypots or similar entities, that asseses the probability or some other non-negative metric, that connection f is harmed by devices on path P. Note that $T(f, P)$ may be symmetric so that it may characterize the effect of f on P, rather than the opposite. Furthermore it may be expressed as the cumulative effect of all the nodes on path P, such as:

$$T(f, P) = \sum_{s \in P} T(f, s),\ \mathit{or}\ T(f, P) = \max\{T(f, s) :\ s \in P\}. \tag{2}$$

$J(f, P)$ is the energy consumed per packet by flow f by devices along path P, which can be computed from the power consumption and traffic rate, as follows:

$$J(f, P) = \sum_{s \in P} \frac{\Pi(s, \lambda(s))}{\lambda(s)}, \tag{3}$$

where $\Pi(s)$ is the power consumption when switch or node s carries the traffic rate $\lambda(s)$ while:

$$\lambda(s) = \sum_{f \in F} \sum_{s \in f} \lambda(f), \tag{4}$$

and $\lambda(f)$ is the traffic rate of connection f, and $F = \{f\}$ is the set of all active connections.

2.2 RNN Based Routing for Path Control

The approach taken here is to use the Cognitive Packet Network (CPN) idea [14,15,23], so as to store inside the SDN Controller a "good" or near-optimal path $P(f)$ for flow $f = (c, e)$ from client c to edge device e that minimizes $G(f, P(f))$. Thus, rather than calculate ex-nihilo for each upcoming connection $f = (c, e)$ the path $P(f)$, we follow the CPN approach that maintains for each router or switch (i.e. node) s, a Random Neural Network [12] that computes the best "next hop" from s to $s'(s, e))$, where $s'(s, e)$ is the node to which s is connected and that minimizes $G((s, e), P(s, e))$.

Since our study is focused on the IoT where the real-time operation is crucial, the path link latencies were chosen as the key QoS metric. Since a SDN controller within its standard means has no direct way to measure the latency on the links and paths, Cognitive Packets (CP) were employed as described in [24] were described for this purpose. CPs have also been employed in SDN networks previously [9,33,34], but the concept of the Cognitive Network Map (CNM) was extended with all necessary data within single data structure.

2.3 Energy

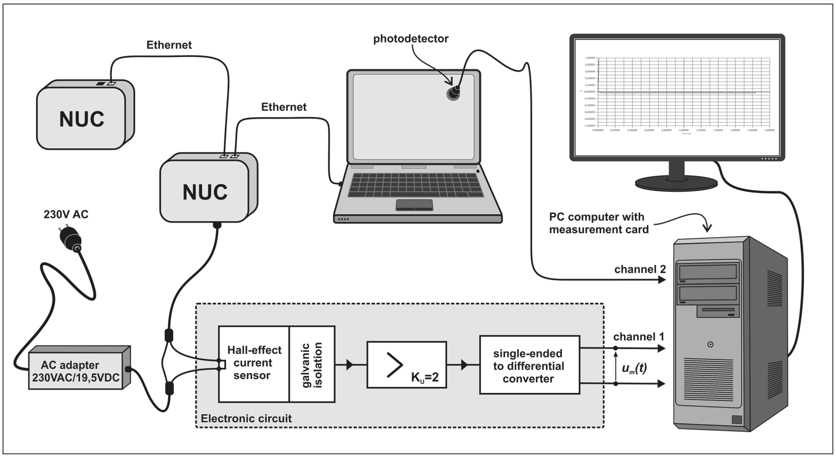


Fig. 1. Measurement circuit for power versus traffic characteristics.

Most network devices do not have the ability to directly measure energy during operation. However, since each network packet handled needs to be processed and transmitted, it is obvious that the amount of energy consumed during operation of a network switch depends on the traffic intensity. The energy characteristic reflecting the amount of energy in Watts [W] depending on the amount of network traffic passing through the switch is, on the one hand, easy to measure in the laboratory, and on the other hand - during operation in the real system - gives the SDN controller, knowing the current throughput of the node, a sufficiently precise answer to the question "How much energy does the network switch consume at this moment".

The SDN switches used in our experiments are Intel NUC devices [26] that run Open vSwitch [29]. Our approach, however, is universal in the sense that it can be applied to any network switch or router.

The laboratory setup used for the measurements if the power drawn during data transfer is presented in Fig. 1. After setting of the traffic level given in Mb/s the energy measurement was done. The traffic was generated and received by workstations connected to the NUC device. The experiment was carried out for successive for increasing traffic levels as shown in Fig. 2.

The electronic circuit which is used to condition the signal obtained from a sensor which measures the current, is based on precision operational amplifiers. The Hall effect-based current sensor ACS712-05 (0–5A current range) is galvanically isolated from the copper conduction path, integrated into the IC, which is used to pass the measured current. This path was connected in series with the supply wire on the constant DC voltage side at $U_{DC} = 19.5V$, of the AC adapter used for the NUC's as shown in Fig. 1. The output signal from the sensor is amplified in a single-ended amplifier and then converted to the differential form. The instantaneous value of the measured power can then be found from the following relationship:

$$P = U_{DC}.i = U_{DC}\frac{U_m}{k_u S} = AU_m,\ in\ Watts\ , \tag{5}$$

where $S = 185mV/A$ is the sensitivity of the current sensor, and $A = U_{DC}/(k_u S) = 520.9A$ is a constant with $k_u = 2$ which is related to the instrumentation, and U_m is the measured output voltage of the single-sided differential converter shown at "channel 1" of Fig. 1, which results from the Hall-effect measurement of the NUC input current.

To reduce the effect of noise and interference, thirty separate measurements were repeated for the power consumption as a function of incoming and outgoing traffic, and the results are summarised in Fig. 2. Then we extracted the difference of the energy consumption between the basic level for zero traffic and the value for a given traffic level, and the increase of energy consumption per traffic volume in Mb is presented in Fig. 3.

2.4 Security

The level of trust in a given flow, and therefore in the device that generates it, can be assessed using external entities. Within the network, nodes and devices with higher and lower sensitivity may be defined. For example, the failure of some nodes has a greater impact on the operation of the entire network than in the case of other nodes, and attacking such a node will cause more damage than otherwise. Security-aware routing aims to direct suspicious traffic away from

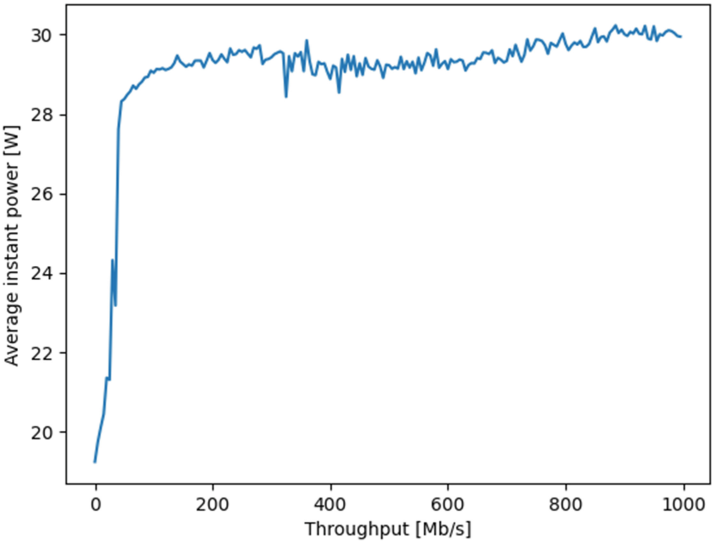


Fig. 2. The dependence between the instantaneous power consumption and traffic load of the Intel NUC when used as a switch or router.

vulnerable nodes, if possible. Trust assessing entities can be Attack Detectors or Honeypots, e.g. [17,30]. We employed SYN attack detector presented in [7].

3 Experiments and Results

The experiments we performed were done in the IITiS laboratory. The test network consisted of seven NUC devices working as SDN switches, plus SDN controller, client machines and attack detector. The basic topology of the network is presented in Fig. 4

For clarity of results presentation, and in order to concisely present the different possibilities of our solution, two separate experiments were performed, however the basic network configuration remained the same. The course and results of the experiments follows.

3.1 Point-to-Point Transmission in Insecure Environment

The aim of the experiment was to reflect the situation of point-to-point communication in the situation of an attack. As presented in Fig. 6, point-to-point communication from c_1 to c_6 client devices was established and put under observation. In this experiment energy efficiency was not taken into consideration, to avoid too many factors influencing the results, making it hard to separate the influence of each of them on the final results.

The experiment had three steps:

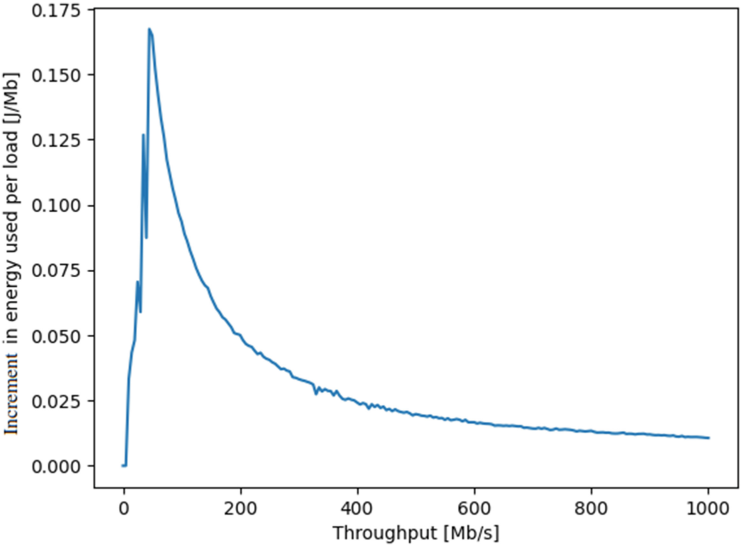


Fig. 3. The energy used per Mb in the function of switch load.

- Normal communication from c_1 to c_6
- QoS deterioration on the link c_1–c_4
- Security problem detected – the need to bypass sensitive nodes c_3 and c_5

The measurement included latency on the path c_1 to c_6. System reaction to changing conditions can be easily observed in the Fig. 5. After some time needed for the neural network to test various conditions and possibilities the path which is both fast and secure was found. The network configurations in particular steps are presented in Figs. 6, 8 and the final on in 8

3.2 Energy-efficient Access to the Edge

The final topology of the second experiment is presented in Fig. 9. It include 24 client devices (implemented as virtual machines) and seven edge services. Every switch was accompanied by the separate service instance. The energy characteristics is taken into account, as well as total time of request handling by the Edge services. The total handling time included: time of client-to-service communication t_{cs}, request handling in the server t_r, time of service-to-client communication t_{sc}. The second component of the goal function was energy efficiency, and energy characteristics from the Fig. 3 was loaded into SDN controller for readouts of energy usage based on traffic in each switch. The RNN decision engine was used for path-and-service choice (Fig. 7).

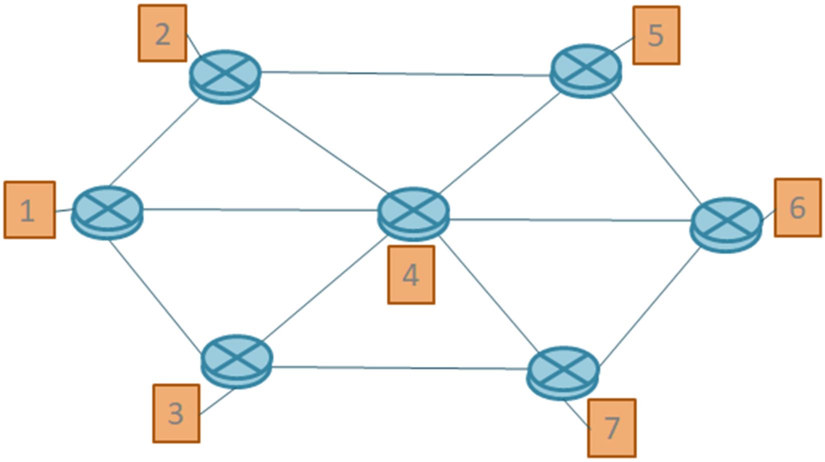


Fig. 4. The configuration of experimental test-bed

The course of the experiment included loading the network with heavy traffic of stress-test type, as such a load was best to show differences in energy usage. Seven steps of experiments were performed, in every step the total load in the network was increased by 1 Gb/s. In the first run only QoS optimisation was performed as a reference result, then both QoS and Energy components were included into the Goal functions. The results, presented in Figs. 10 and 11, show positive influence of the latter version of Goal function on the total energy consumption. with minor effect on QoS.

4 Conclusions

The paper presents the possibilities of using modern tools from the field of Artificial Intelligence (AI) and Machine Learning (ML) to control the operation of computer networks. It has been shown that theoretical capabilities of RNNs can be translated into practical applications, and appropriately constructed goal functions perform complex routing based on several criteria simultaneously.

Among the criteria tested experimentally are the possibilities of increasing the security and reducing the energy consumption of the IT infrastructure, which are very relevant for today's IT systems. These very promising ideas have been tested in several experiments which demonstrate their practical value in the framework of Software Defined Networks.

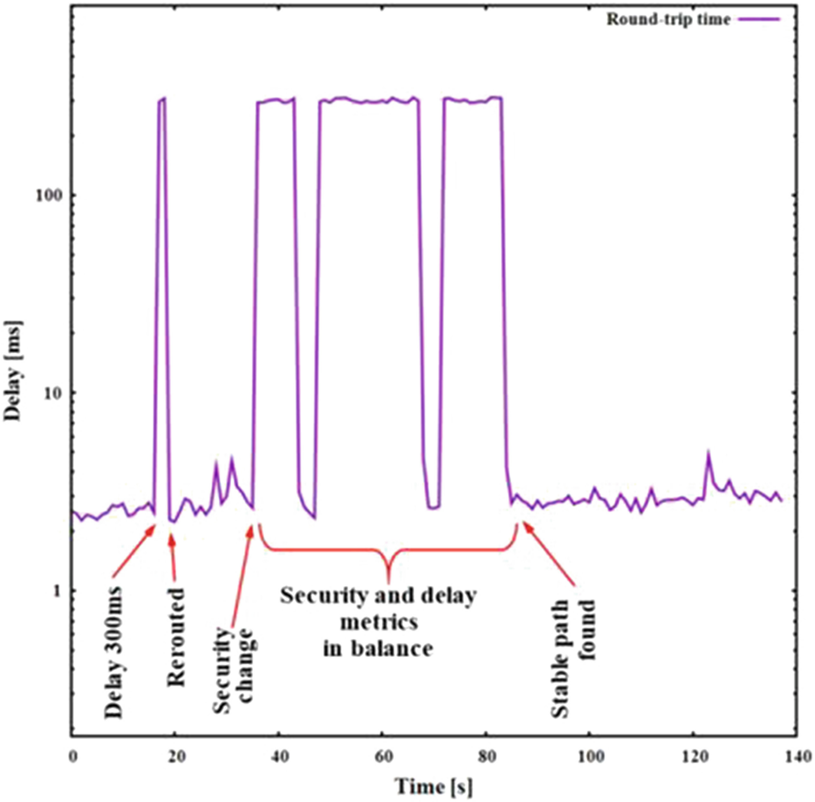


Fig. 5. The delay in time between clients 1 and 6

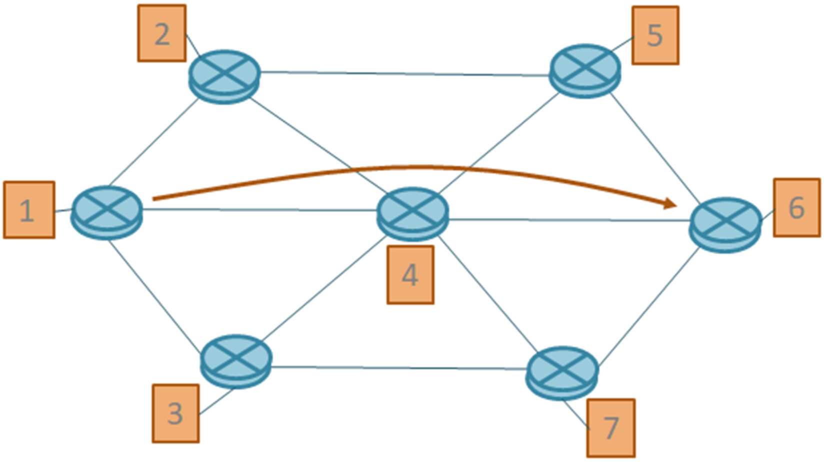


Fig. 6. The c_1–c_6 path configuration – stage 1

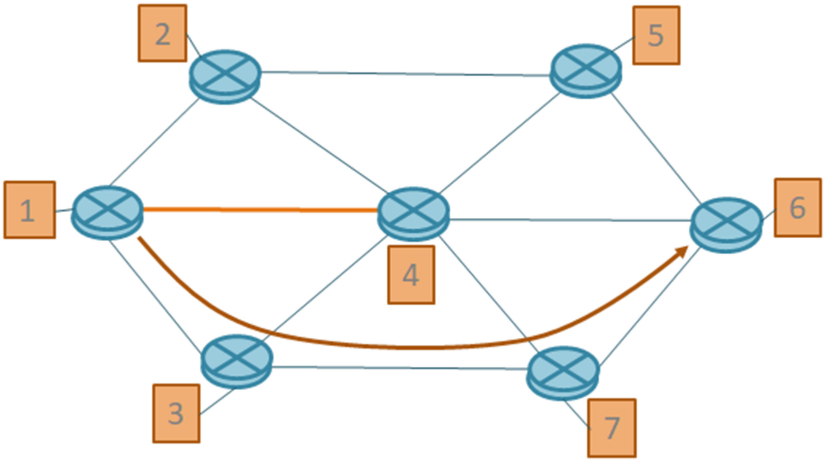


Fig. 7. The c_1–c_6 path configuration – stage 2

References

1. Andrae, A.S.G., Edler, T.: On global electricity usage of communication technology: trends to 2030. Challenges **6**(1), 117–157 (2015)
2. Aytaç, S., Ermiş, O., Çağlayan, M.U., Alagöz, F.: Authenticated quality of service aware routing in software defined networks. In: Zemmari, A., Mosbah, M., Cuppens-Boulahia, N., Cuppens, F. (eds.) CRiSIS 2018. LNCS, vol. 11391, pp. 110–127. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-12143-3_10
3. Berl, A., et al.: Energy-efficient cloud computing. Comput. J. **53**(7), 1045–1051 (2010)
4. Brun, O., Wang, L., Gelenbe, E.: Big data for autonomic intercontinental overlays. IEEE J. Sel. Areas Commun. **34**(3), 575–583 (2016)
5. Brun, O., Yin, Y., Gelenbe, E.: Deep learning with dense random neural network for detecting attacks against IoT-connected home environments. Procedia Comput. Sci. **134**, 458–463 (2018)
6. Diamantopoulos, S., et al.: Secure cross-border exchange of health related data: the KONFIDO approach. In: Montella, R., Ciaramella, A., Fortino, G., Guerrieri, A., Liotta, A. (eds.) IDCS 2019. LNCS, vol. 11874, pp. 318–327. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34914-1_30
7. Evmorfos, S., Vlachodimitropoulos, G., Bakalos, N., Gelenbe, E.: Neural network architectures for the detection of SYN flood attacks in IoT systems. In: PETRA 2020: Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments. Association for Computing Machinery, New York, NY, United States, Corfu, Greece, June 2020. https://doi.org/10.1145/3389189.3398000
8. Francois, F., Gelenbe, E.: Optimizing secure SDN-enabled inter-data centre overlay networks through cognitive routing. In: 2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 283–288. IEEE (2016)

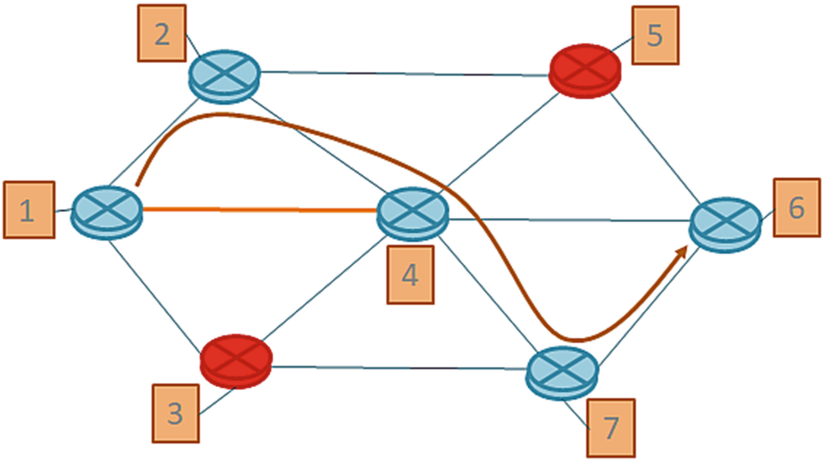


Fig. 8. The c_1–c_6 path configuration – stage 3

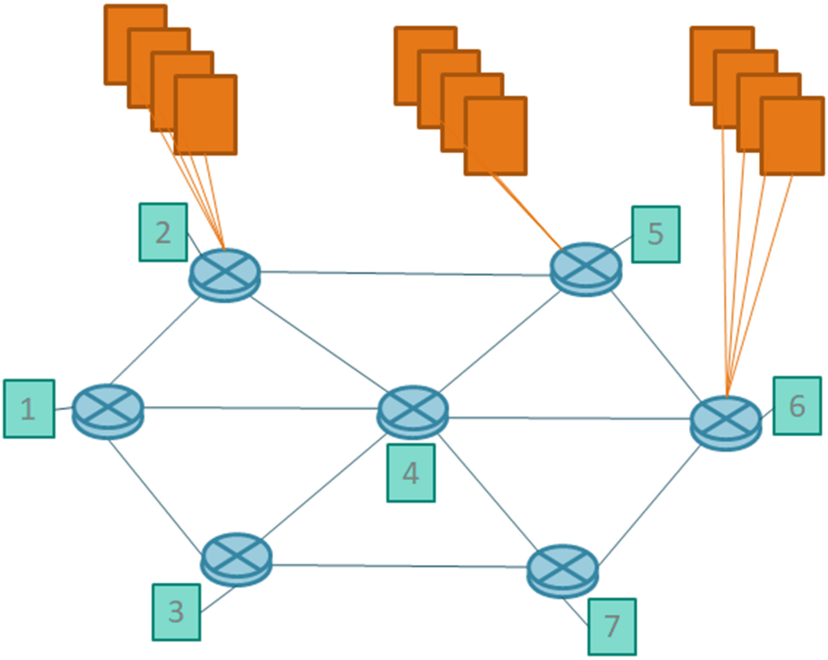


Fig. 9. Configuration of the Edge services experiment

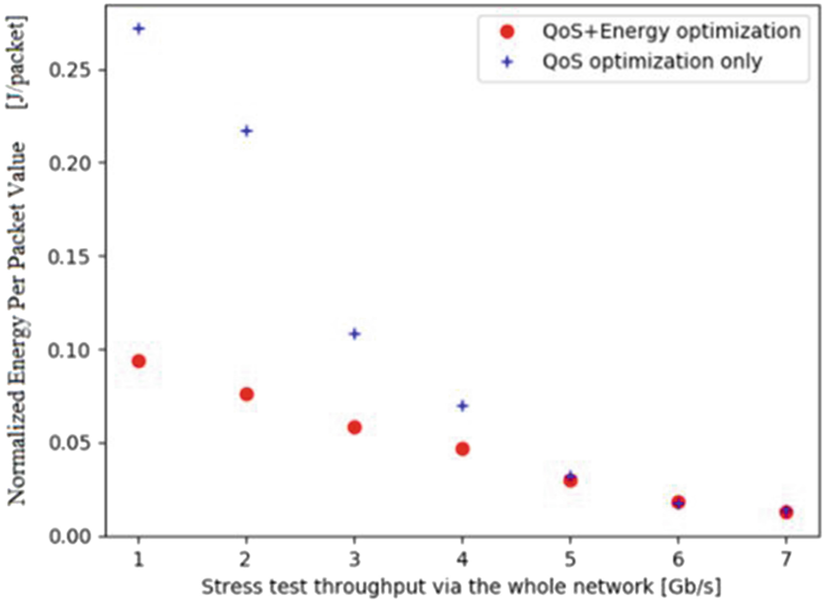


Fig. 10. Average Energy [J]/packet during stress test

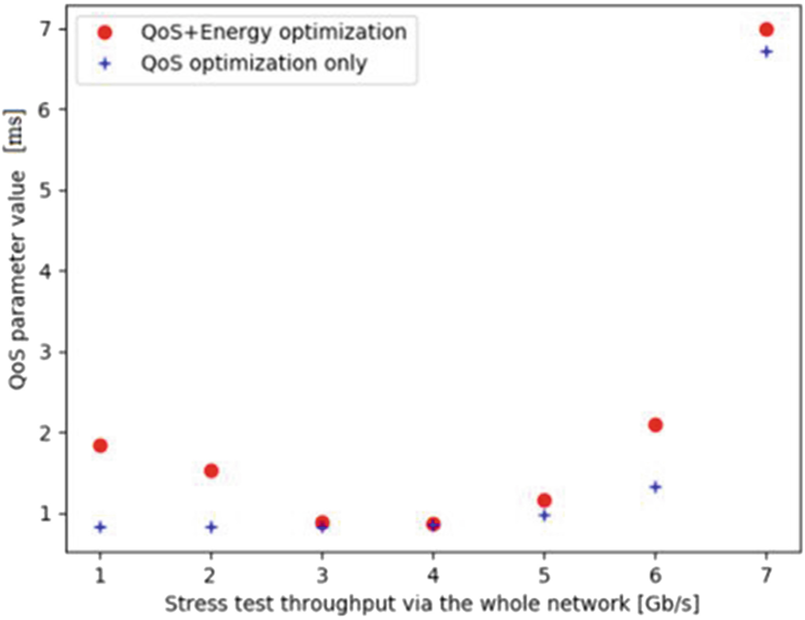


Fig. 11. Average QoS (delay [ms]) during stress test

9. François, F., Gelenbe, E.: Towards a cognitive routing engine for software defined networks. In: 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, May 22–27, pp. 1–6 (2016). https://doi.org/10.1109/ICC.2016.7511138
10. Frohlich, P., Gelenbe, E., Nowak, M.P.: Smart SDN management of fog services. In: GIOTS 2020: Global IoT Summit 2020, IEEE Communications Society, 1–5 June 2020, Dubin, Ireland. TechRxiv (2020)
11. Fröhlich, P., Gelenbe, E., Nowak, M.P.: Smart SDN management of fog services. In: 2020 Global Internet of Things Summit (GIoTS), pp. 1–6 (2020). https://doi.org/10.1109/GIOTS49054.2020.9119542
12. Gelenbe, E.: Random neural networks with negative and positive signals and product form solution. Neural Comput. **1**(4), 502–510 (1989)
13. Gelenbe, E.: Learning in the recurrent random neural network. Neural Comput. **5**(1), 154–164 (1993)
14. Gelenbe, E.: Cognitive Packet Network. US Patent US6804201B1 (2004)
15. Gelenbe, E.: Steps toward self-aware networks. Commun. ACM **52**(7), 66–75 (2009)
16. Gelenbe, E., Caseau, Y.: The impact of information technology on energy consumption and carbon emissions. Ubiquity **2015**, 1–15 (2015)
17. Gelenbe, E., Fröhlich, P., Nowak, M., Papadopoulos, S., Protogerou, A., Drosou, A., Tzovaras, D.: IoT network attack detection and mitigation. In: 9th Mediterranean Conference on Embedded Computing (MECO), Budva, Montenegro, pp. 1–6 (2020). https://doi.org/10.1109/MECO49872.2020.9134241
18. Gelenbe, E., Lent, R.: Energy-QoS trade-offs in mobile service selection. Future Internet **5**(2), 128–139 (2013). https://doi.org/10.3390/fi5020128
19. Gelenbe, E., Loukas, G.: A self-aware approach to denial of service defence. Comput. Netw. **51**(5), 1299–1314 (2007)
20. Gelenbe, E., Mitrani, I.: Analysis and Synthesis of Computer Systems, vol. 4. World Scientific, London (2010)
21. Gelenbe, E., Pavloski, M.: Performance of a security control scheme for a health data exchange system. In: IEEE International Black Sea Conference on Communications and Networking 26–29 May 2020 // Virtual Conference, pp. 1–6. IEEE (2020)
22. Gelenbe, E., Siavvas, M.: Minimizing energy and computation in long-running software. Appl. Sci. **11**(3), 1169 (2021)
23. Gelenbe, E., Xu, Z., Seref, E.: Cognitive packet networks. In: Proceedings 11th International Conference on Tools with Artificial Intelligence, Chicago, IL, USA, pp. 47–54 (1999). https://doi.org/10.1109/TAI.1999.809765
24. Gelenbe, E., Xu, Z., Seref, E.: Cognitive packet networks. In: Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, pp. 47. ICTAI 1999, IEEE Computer Society, USA (1999)
25. Huang, X., Cheng, S., Cao, K., Cong, P., Wei, T., Hu, S.: A survey of deployment solutions and optimization strategies for hybrid SDN networks. IEEE Commun. Surv. Tutorials **21**(2), 1483–1507 (2019). https://doi.org/10.1109/COMST.2018.2871061
26. Intel: NUC - Small Form Factor Mini PC. https://en.wikipedia.org/wiki/Next-Unit-of-Computing (2021)
27. Kehagias, D., Jankovic, M., Siavvas, M., Gelenbe, E.: Investigating the interaction between energy consumption, quality of service, reliability, security, and maintainability of computer systems and networks. SN Comput. Sci **2**(1), 1–6 (2021)
28. Kim, C., Kameda, H.: An algorithm for optimal static load balancing in distributed computer systems. IEEE Trans. Comput. **41**, 381–384 (1992)

29. McKeown, N., et al.: OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Comput. Commun. Rev. **38**(2), 69–74 (2008)
30. Nakip, M., Gelenbe, E.: MIRAI botnet attack detection with auto-associative dense random neural network. In: 2021 IEEE Global Communications Conference. Barcelona, Spain, December 2021
31. Nalin, M., et al.: The European cross-border health data exchange roadmap: case study in the Italian setting. J. Biomed. Inf. **94**, 103183 (2019)
32. Natsiavas, P., et al.: Developing an infrastructure for secure patient summary exchange in the EU context: Lessons learned from the konfido project. Health Inf. J. **27**(2), 14604582211021460 (2021)
33. Nowak, M., Nowak, S., Domanska, J.: Cognitive routing for improvement of IoT security. In: Proceedings of IEEE International Conference on Fog Computing ICFC, Prague (2019). https://doi.org/10.13140/RG.2.2.28667.36648
34. Nowak, M., Nowak, S., Domańska, J., Czachórski, T.: Cognitive packet networks for the secure internet of things. In: Global IoT Summit (GIoTS). Aarhus, Denmark (2019)
35. Pernici, B., Aiello, M., Vom Brocke, J., Donnellan, B., Gelenbe, E., Kretsis, M.: What is can do for environmental sustainability: a report from caise'11 panel on green and sustainable is. Commun. Assoc. Inf. Syst. **30**(1), 18 (2012)
36. Rawat, D.B., Lenkala, S.R.: Software defined networking architecture, security and energy efficiency: a survey. IEEE Commun. Surv. Tutorials **19**(1), 325–346 (2017). https://doi.org/10.1109/COMST.2016.2618874
37. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. 2nd Ed. MIT Press, Cambridge (2018)
38. Tian, W., Zhao, Y., Zhong, Y., Xu, M., Jing, C.: A dynamic and integrated load-balancing scheduling algorithm for cloud datacenters. In: Proceedings of IEEE International Conference on Cloud Computing and Intelligence Systems, pp. 311–315 (2011)
39. Topcuouglu, H., Hariri, S., Wu, M.Y.: Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Trans. Parallel Distrib. Syst. **13**(3), 260–274 (2002)
40. Wang, L., Brun, O., Gelenbe, E.: Adaptive workload distribution for local and remote clouds. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 3984–3988 (2016)
41. Zhang, Z., Zhang, X.: A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation. In: Proceedings of 2nd International Conference Industrial Mechatronics Automation, vol. 2, pp. 240–243 (2010)
42. Zhu, X., Qin, X., Qiu, M.: Qos-aware fault-tolerant scheduling for real-time tasks on heterogeneous clusters. IEEE Trans. Comput. **60**(6), 800–812 (2011)

Mitigating the Massive Access Problem in the Internet of Things

Erol Gelenbe[1,2,3,4](✉), Mert Nakıp[1], Dariusz Marek[5], and Tadeusz Czachorski[1]

[1] Institute of Theoretical and Applied Informatics Polish Academy of Sciences,
44-100 Gliwice, Poland
{seg,mnakip,tadek}@iitis.pl
[2] Yaşar University, Bornova, İzmir, Turkey
[3] CNRS Lab. I3S Université, Côte d'Azur, Nice, France
[4] CNRS Lab. Abraham de Moivre, London, UK
[5] Faculty of Automatic Control, Electronics and Computer Science,
Silesian University of Technology, 44-100 Gliwice, Poland
dariusz.marek@polsl.pl

Abstract. The traffic from the large number of IoT devices connected to the IoT is a source of congestion known as the Massive Access Problem (MAP), that results in packet losses, delays and missed deadlines for real-time data. This paper reviews the literature on MAP and summarizes recent results on two approaches that have been designed to mitigate MAP. One approach is based on randomizing the packet arrival instants to IoT gateways within a given time interval that is chosen so that packet arrivals do not exceed their deadlines, but also so that they do not constitute bulk arrivals. The second approach is a novel traffic shaping policy named the Quasi-Deterministic-Transmission-Policy (QDTP) which has been proved to drastically reduce queue formation at the receiving gateway by delaying packet departures from the IoT devices in a judicious manner. Both analytical and experimental results are summarized, that describe the benefits of these techniques.

Keywords: Internet of Things (IoT) · IoT Gateways · Massive Access Problem · Diffusion approximation · Trace driven simulations · Quasi-Deterministic Transmission Policy

1 Introduction

The number of Internet of Things (IoT) devices is increasing rapidly with the increasing needs of smart cities, healthcare applications, autonomous systems, and smart vehicles [6,7,12,40], causing the overload of communication channels and gateways [27]. This results in the Massive Access Problem (MAP) where high latency and queue lengths can lead to packet loss and deadline violations. In addition, congestion can lead to increased energy consumption at IoT devices and gateways due to repeated requests for access and increased processing times

E. Gelenbe et al. (Eds.): EuroCybersec 2021, CCIS 1596, pp. 118–132, 2022.
https://doi.org/10.1007/978-3-031-09357-9_10

[2,15], thus contributing to the worldwide increase in energy consumption for ICT [14].

Thus, substantial work over the last several years [3,11,28–30,34,35,39,46–51,55,56] attempts to solve MAP in various ways.

In this paper, we first briefly review methodologies and results with regard to reactive or proactive (predictive) solutions that can mitigate MAP. Then, we summarize two recent research avenues: Randomization of Transmission Times [36] and a novel traffic shaping policy – the Quasi-Deterministic-Transmission-Policy (QDTP) [18,19,25]. We illustrate the results these approaches offer via analytical techniques and trace driven simulations using a publicly available dataset of up to 6400 IoT devices [1] with different deadline constraints.

The remainder of this paper is organized as follows. Section 2 reviews recent research focusing on MAP. Section 3 summarizes two recent studies on MAP based on analytical and experimental results. Finally in Sect. 4 our main conclusions are presented.

2 Review of Prior Work on MAP

This section reviews the prior work on MAP in two categories as reactive solutions and predictive/proactive solutions. Early research addressed MAP by reducing congestion through adaptive Random Re-Routing (RRR) [20–22] which improves the QoS of a sensor network by dynamically changing packet routes when congestion is detected. In related work [42], an information theoretic technique selectively reduces the amount of traffic by increasing transmission efficiency, and improves the QoS in sensor networks.

More recent work has proposed solutions to MAP, assuming that IoT traffic is generated at random, and using approaches mostly based on Access Class Barring (ACB). In [35] ACB is enhanced by using Markov chains to model the status of preambles and to forecast active devices, while other work [30] developed the recursive ACB algorithm based on instantaneous detection of idle preambles. Recent work [28] also developed recursive ACB which adapts the probability that a device sends a preamble based on estimating the number of active IoT devices. The performance of ACB has been analyzed under different parameters for Machine-to-Machine (M2M) communications [55], and enhanced by Reinforcement Learning (RL) to select its parameter (i.e. barring rate) with respect to network conditions [56]. In [29] deep RL techniques are proposed to maximize the number of devices that successfully access the medium without collisions.

In [34] an access scheme for M2M communication that clusters machine-type devices according to their requirements and locations is used. The Non-Orthogonal Multiple Access (NOMA) based technique is presented for networks with a massive number of devices in [51]. Moreover, in order to address MAP, in [3] a collision detection based random access technique is developed, and in [49] a hybrid technique that combines slotted-Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) and Time-Division Multiple Access (TDMA) schemes are suggested.

2.1 Proactive Solutions

IoT devices are quite simple and cannot easily be coordinated for timing or scheduling [24] based on distributed control [8,10]. Thus recent experimental results [37,38] show that machine learning techniques can be used to predict IoT traffic generated by individual devices, and other work [11,39,46–48,50] designs proactive/predictive access schemes that determine the transmission times from IoT devices based on such predictions to mitigate the MAP.

Other work [31,45,48] has developed proactive access schemes for Machine-to-Human (M2H) or Human-to-Machine (H2M) traffic, while earlier research [43,44] has focused on Human-to-Human (H2H) traffic. In [43] a schedule-based protocol an expert system is used to determine schedules that minimize delay and maximize channel utilization. In [44], forecasts of data rates of individual applications are used to schedule channel scheduling, and network load has also been balanced based on the forecast of the total load of all machine-type devices [31]. To mitigate the latency bottleneck due to the contention in optical or wireless networks [48], the proactive allocation of bandwidth to transmissions of packet bursts based on Artificial Neural Network (ANN) forecasts is studied, while in [45] the prediction of network throughput for better Quality of Experience (QoE) is investigated.

Fast Uplink Grant (FUG) is one of the predictive access schemes presented in 3GPP Release 14 to provide predetermined uplink allocations for IoT devices [4,5,52]. In [50], IoT packet transmissions have been modelled with a binary Markov process, while in [11] a FUG allocation technique was developed by combining Support Vector Machines (SVM) and Long-Short Term Memory (LSTM) neural networks.

Another trend [39,46,47] on predictive access schemes addresses MAP by using Joint Forecasting Scheduling (JFS), and in [39] JFS was proposed to schedule transmissions based on forecasts of generation times and sizes of bursts. JFS can be recursively enhanced with a Multi-Scale Algorithm (MSA), where the performance of JFS and the length of scheduling horizon, are significantly increased [46] (over 96% throughput) for 6400 devices with variable latency constraints. However, the computational requirements of MSA are very high and in [47] a scheduling heuristic is used to determine JFS transmission times for using multiple frequency channels.

Most recently the Randomization of Generation Times (RGT) preprocessing algorithm [36] is shown to significantly improve the performance of scheduling heuristics with very low computational cost for large numbers of IoT devices. Also, the Quasi-Deterministic Transmission Policy (QDTP) traffic shaping approach [19,25] has been shown, using queueing theory, diffusion approximations [16] and trace driven simulations, to mitigate the MAP by drastically reducing the waiting time at gateways. This research has shown that RGT and QDTP can mitigate MAP with very low computational requirements for up to 6400 IoT devices. In the remainder of this article, we shall outline how these two avenues of recent research can alleviate MAP.

3 Mitigating MAP Using Queueing Theory and Diffusion Approximations

We will summarize together the results of two recent articles [19,36], which offer solutions to reduce MAP. To this end, we first present the analysis in [19] of the probability that the deadline of an IoT packet is met, providing a basis for access policies in IoT networks. Then we review the RGT [36] and QDTP [19] algorithms and their performance.

In [19], the collection of IoT devices that generate packets, the communication channel, and the receiving gateway are represented as two cascaded queues [9, 32,41,53]:

- The first queue translates the generation instant at the IoT device for the j-th packet r_j, into its transmission instant t_j, and
- The second queue starts with the transmission instants t_j that feeds directly into the IoT gateway where the j-th packet is served in FIFO (First-In-First-Out) order with a service duration p_j.
- Note that in this case, the transmission delay within the communication channel is taken to be zero, i.e. it is assumed to be small enough to be as compared to p_j and to the durations between the other successive instants, so that the packet leaving the IoT device at time t_j arrives at the gateway at the same time instant.

Let the IoT device generats traffic in bursts of bits that are sent at the same time instant, where burst j is generated at discrete time slot r_j and should be recieved by d_j. That is, we assume that there is a deadline Δ_j for each burst j beyond which j is of “no value”. Thus, the burst j (which can also be considered as a packet) must arrive at the receiver gateway by $d_j = r_j + \Delta_j$. Furthermore, the packets of various IoT devices are processed in time ordered fashion in First-In-First-Out (FIFO) order, and p_j be the “service time” during which the receiving gateway is occupied by packet j.

3.1 The Probability of Meeting Deadlines

Let the j-th packet sent from any of the IoT devices, enumerated in time order (i.e. the $j - th$ packet is generated before the $j + 1$-th packet), be transmitted from its IoT device exactly at the instant r_j when it is generated. Then the time spent between the generation time r_j and the time when it starts being processed at the gateway, i.e. its total waiting time, is denoted by W_j, and is given by Lindley’s recursive equation [32,53]:

$$W_{j+1} = [W_j + p_j - r_{j+1} + r_j]^+ \ , \ j = 0, 1, 2, \ . \tag{1}$$

where $r_0 = 0$. Note that the conventional notation $[X]^+$, for a real number X, means that $[X]^+ = 0$ when $X < 0$, and $[X]^+ = X$ if $X \geq 0$.

Assuming that the generation times coincide with the transmission times, if the sequence generation and service times, and deadlines $\{r_j,\ p_j,\ \Delta_j\}_{j\geq 0}$

constitute a stationary random process, the probability Π_j that the packet j does not meet its deadline is given by:

$$\Pi_j = Prob[R_j = W_j + p_j > r_j + \Delta_j],\ and\ \Pi = \lim_{j \to \infty} \Pi_j, \tag{2}$$

where R_j is known as the response time.

Since the focus of the work in [19,36] is on selecting the transmission instant of each packet j, denoted by t_j, to minimize Π_j under each traffic load of the network, W_j will be replaced by a total end-to-end delay V_j to each packet, where D_j is a scheduling delay imposed to each successive packet, and V_j includes the delay at the IoT device plus the delay at the gateway:

$$V_j = W_j^a + W_j^b,\ j \geq 0,\ V_0 = W_0^a = W_0^b = 0,\ and \tag{3}$$

$$W_{j+1}^a = [W_j^a + D_j - (r_{j+1} - r_j)]^+,\ t_j = r_j + W_j^a, \tag{4}$$

$$W_{j+1}^b = [W_j^b + p_j - (t_{j+1} - t_j)]^+. \tag{5}$$

In other words, we impose an initial scheduling delay D_j to each successive packet, and then consider the resulting effect on the transmission instant t_j and on the resulting delay at the IoT device W_j^a followed by the delay at the gateway W_j^b. These matters are analyzed in detail in [25], wth a resulting effect on the probability of missing the deadlines:

$$\Pi_j^* = Prob[R_j^* = V_j + p_j > r_j + \Delta_j],\ and\ \Pi^* = \lim_{j \to \infty} \Pi_j, \tag{6}$$

when $\{r_j,\ p_j,\ D_j,\ \Delta_j\}_{j \geq 0}$ constitute a stationary random process.

3.2 Interarrival and Service Time Statistics

Next, in order to compute the interarrival and service time statistics, the approach in [19] assumes that the p_j's of all packets are independent random variables with the same distribution whose mean is $E[P]$ and its SCV is C_B^2. Moreover, λ denotes the interarrival rate of packets, such that $\lambda = E[r_{j+1} - r_j])^{-1}$. It has also been assumed that the value of λ increases with the number of IoT devices M that are connected to the IoT gateway, and the system will operate under variable λ (or M) but under stable conditions, i.e. $\lambda E[P] < 1$. Also, let C_A^2 denote the SCV of interarrival times of packets. Then, C_B^2 and C_A^2 are respectively defined as

$$C_B^2 = \frac{E[P^2]}{(E[P])^2} - 1,\ \text{and}\ C_A^2 = \frac{E[(r_{j+1} - r_j)^2]}{(E[r_{j+1} - r_j])^2} - 1. \tag{7}$$

3.3 Using the Diffusion Approximation:

Last, the diffusion approximation [13,33] has been used in order to determine the probability that $R_j \leq \Delta_j$, denoted by $F_R(\Delta)$, where $\Delta_j = \Delta$ which is constant for all packets of all IoT devices. Then, the probability of missing deadline $\Pi = 1 - F_R(\Delta)$.

Subsequently, using the diffusion approximations [19] one obtains the probability density function of the response time and the probability of missing deadline as

$$f_R(t) = \int_0^\infty \frac{x}{\sqrt{2\pi\alpha t^3}} e^{-\frac{(x+\beta t)^2}{2\alpha t}} f(x)dx, \text{ then } \Pi = 1 - \int_0^\Delta f_R(\tau)d\tau, \quad (8)$$

where $\beta = \lambda - \mu, \quad \alpha = \lambda C_A^2 + \mu C_B^2$, $\mu = 1/E[P]$, and $f(x)$ is yielded by the diffusion model [13].

3.4 Numerical Results Concerning the Diffusion Analysis

Now, the results concerning a publicly available IoT traffic dataset [1] present how Π varies with C_A^2 and Δ.

First, Fig. 1 displays $log_{10}(\Pi)$ for $\lambda = 0.8$ and $C_B^2 = 1$ and different values of Δ and C_A^2. Note that the minimum and maximum values of C_A^2 for the traffic in the dataset [1] which are 1.6 and 2.18, and the approximate value of C_A^2 for the uniform distribution which will be used for the randomization policy in Sect. 3.5, are shown as vertical bars. The results in this figure show that Π increases with C_A^2 and decreases with Δ when λ remains constant.

Then, Fig. 2 presents the values of Π against increasing number of devices M as well as the corresponding λ and C_A^2. The results show that the measured value of Π, which is the fraction of packets that do not meet their deadlines, increases as M or corresponding λ increases.

In summary, the results in Fig. 1 and Fig. 2 from [19] show that reducing C_A^2 significantly increases the probability that any IoT traffic packet meets its deadline.

3.5 Randomization of Data Generation Times (RGT)

While the results in Sect. 2 show that fast and computationally inexpensive heuristic algorithms are promising for MAP, the work in [36] develops the RGT preprocessing algorithm which relieves the traffic load by distributing the generation times of packets over a scheduling window of duration of T_{sch}.

In RGT, r_j is updated by adding to it an offset which is a uniformly distributed random variable (Recall that $C_A^2 = 1/3$ for uniform distribution which is shown to be significantly lower than the minimum value in the dataset) as

$$r_j^{new} \leftarrow r_j + U[\Delta_j - S_j] \quad (9)$$

where S_j is a *safety delay* that limits the upper bound of r_j^{new} such that $0 \leq S_j \leq \Delta_j$ which indicates the maximum randomization and $S_j = \Delta_j$ indicates that there is no randomization.

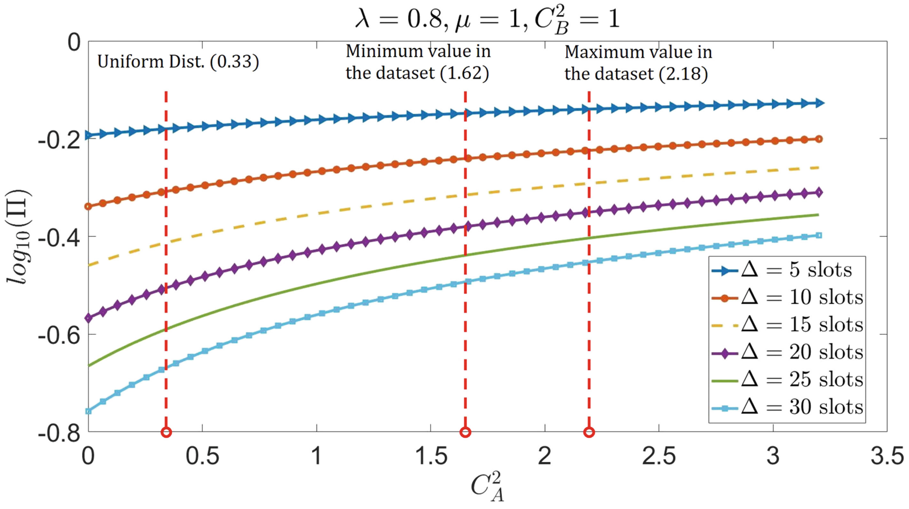


Fig. 1. We show the probability of missing the deadline (y-axis) in logarithmic scale (to the base ten), estimated with the diffusion approximation. We see that it increases significantly as C_A^2 increases and when the deadline Δ measured in slots decreases, for a fixed but high value of the arrival rate $\lambda = 0.8$. The average service rate μ and the SCV of service time C_B^2 are both fixed to 1.

Then, in [36] the value of S_j was selected by using queueing theory [17,23] as:

$$S_j \approx min[\ p_j + \frac{E[P]}{\frac{1}{\rho} - 1},\ \Delta_j\ ], \tag{10}$$

where $E[P]$ is the average processing time, and $\rho = \lambda E[P]$.

3.6 Experimental Results Concerning RGT

We now review the performance evaluation of RGT for two known heuristic scheduling algorithms, Priority based on Average Load (PAL) [39] and non-preemptive version of Earliest Deadline First (EDF) [26], where PAL enhanced with RGT is called R-PAL and EDF enhanced with RGT is called R-EDF. The performance evaluation is performed on the publicly available dataset [1] with respect to each of throughput η and fraction of successfully delivered bursts ζ metrics. Also, the performances of R-PAL and R-EDF are compared with the upper bound performances, where $S_j = \gamma \Delta_j$ and exhaustively search for the value of γ in the range $[0, 1]$ with increments of 0.05 to maximize each of η and ζ.

Accordingly, Fig. 3 displays the comparison of R-PAL and R-EDF with the upper bounds of those as well as PAL and EDF heuristics for η and ζ. The results in this figure show that RGT preprocessing significantly improves the throughput performance of each heuristic while the fraction of successfully delivered bursts

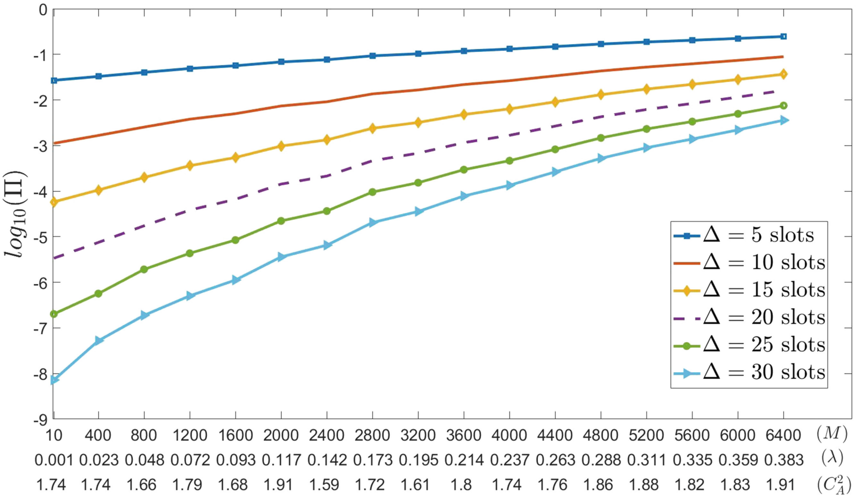


Fig. 2. The probability of missing the deadline (y-axis) in logarithmic scale (to the base ten) estimated with the diffusion approximation, using the traffic statistics of the real dataset of [1], is plotted against the number of IoT devices M (x-axis) that are being used. Note that each value of M corresponds to specific measured values of λ and C_A^2 shown along the $x - axis$.

remains the same. Furthermore, the results in Fig. 3 (top) show that the R-PAL and R-EDF significantly outperform the original versions of the heuristics PAL and EDF for a higher number of devices while both R-PAL and R-EDF are able to achieve almost the same throughput with their upper bounds. On the other hand, in Fig. 3 (bottom), one sees that ζ is almost the same for the enhanced heuristics (R-PAL and R-EDF) and the original heuristics (PAL and EDF).

3.7 The Quasi-Deterministic Transmission Policy (QDTP)

As the diffusion analysis that is discussed in Sect. 3.4 suggests, minimizing the SCV of interarrival times of traffic packets, C_A^2, reduces the probability of missing deadlines of those packets. Accordingly, in [19] a "Quasi-Deterministic Transmission Policy" (QDTP) is developed to minimize the probability of missing deadlines Π by setting almost all of the intertransmission times to a constant

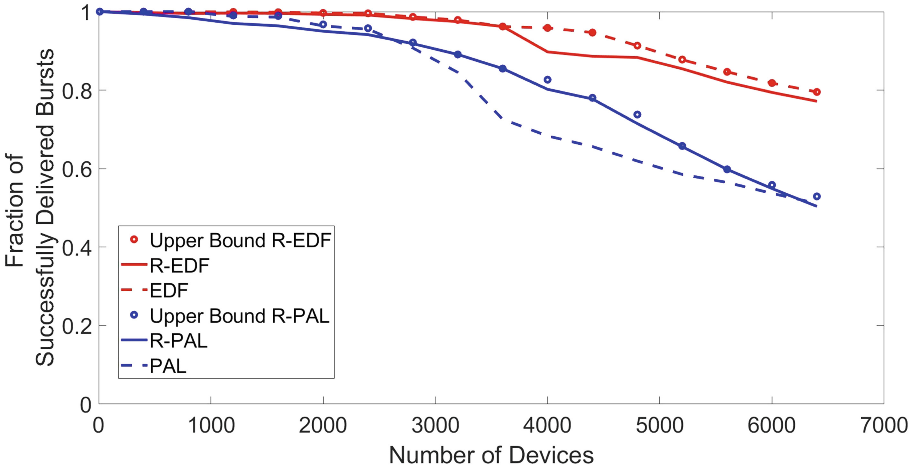


Fig. 3. Comparison of the upper bound R-PAL, R-PAL, PAL and the upper bound R-EDF, R-EDF, EDF algorithms for 12 to 6400 devices with respect to η (top) and ζ (bottom)

D so as to reduce the value of C_A^2. In QDTP whose pseudo-code is presented in Algorithm 1, $D = \frac{1}{\lambda}$, where λ is the interarrival time of burst generation times.

Algorithm 1: Pseudo-code QDTP

```
$n = 1$;
$t_n = a_n$;
for $n \in \{2, \ldots, N\}$ do
    if $a_n \leq t_{n-1} + D$ then
        $t_n = a_{n-1} + D$;
    else
        $t_n = a_n$;
    end
end
```

For the practical application of QDTP, the generation times of the packets must be known in advance, similar to other predictive protocols. While advanced knowledge of the λ value is required, it can be easily calculated based on the creation times of the packages. Also, each IoT device must be informed of the time slot reserved for transmission of a packet created by that device before the start of the reserved slot. For this purpose, information regarding channels reserved for transmission permissions will be sent to devices via the downlink channel. Therefore, the communication channel must be bidirectional or individual IoT devices must also have the ability to detect the channel.

3.8 Experimental Results Concerning QDTP

We now examine the performance evaluation of QDTP on the same dataset [1] used for the evaluation of RGT. On the other hand, the bits are now packetized as IP packets, where the $21 - Byte$ headers (similar to the header size in LoRa-WAN) are followed by the payload, which is the bits in a burst. In addition, considering the effective data transmission rate of LoRa-WAN in free space is 5400 bit/s [54], the transmission time for the average packet length in the dataset is 33.33 ms.

Figure 4 presents the SCV of interarrival times of packets, C_A^2, for both the original generation times in the dataset and the transmission time determined via QDTP. The results in this figure show that C_A^2 is significantly reduced for all values of M when QDTP is used to schedule packet transmissions.

Figure 5 displays the comparison between the performance of QDTP and that of original generation times in the dataset with respect to the base 10 of the (empirically measured) probability Π that the deadline is missed for different values of deadline Δ. In this figure, we see that Π is reduced to practically zero for $\Delta \geq 5$ when QDTP is used. On the other hand, when QDTP is not used (i.e. packets are transmitted when they are generated), Π increases with M and approaches 1.

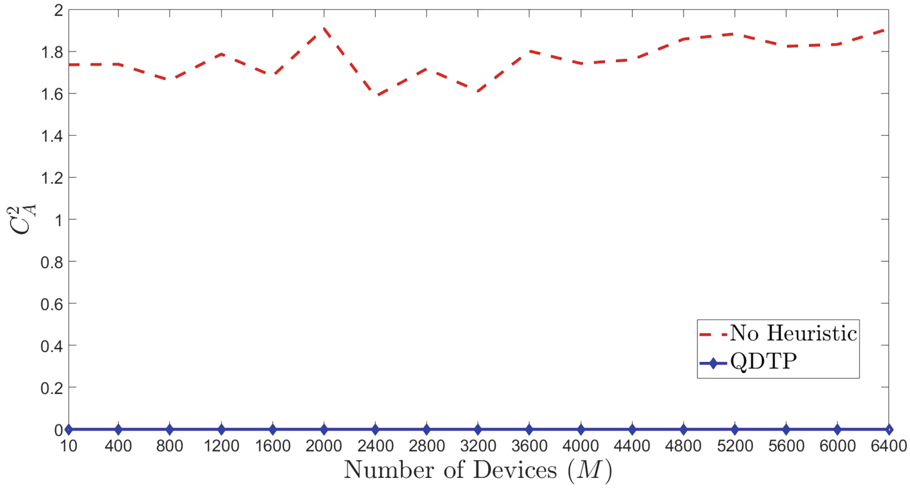


Fig. 4. Measurements of the SCV of interarrival times, both for the raw IoT data from [1], and for the same data using QDTP, for a varying number of active IoT devices M. We observe that QDTP has substantially reduced the empirically measured SCV C_A^2, reducing it to zero for all the distinct numbers of devices M in the dataset of [1].

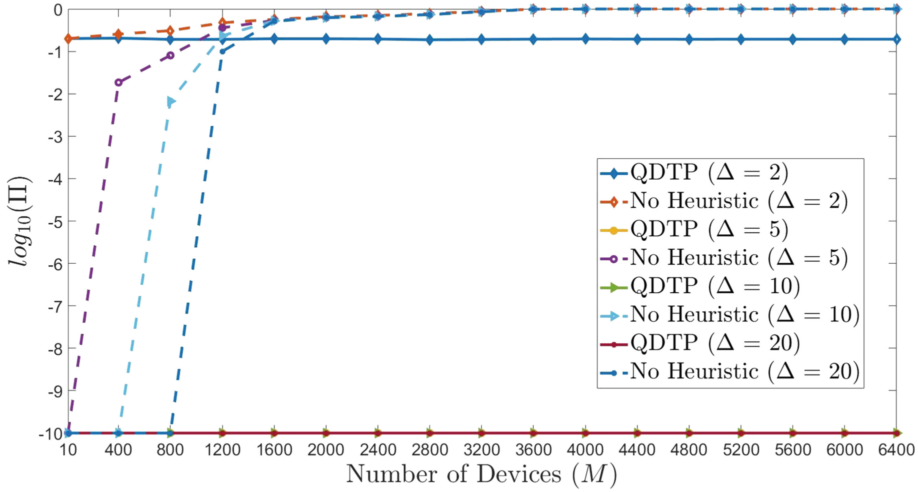


Fig. 5. We compare the logarithm to the base 10 of the (empirically measured) probability Π that the deadline is missed under both the raw dataset of [1] and under the case where the QDTP is used with the same dataset for values of the deadline $\Delta \in \{2, 5, 10, 20\}$.

4 Conclusions

The MAP, which can occur when a massive number of devices attempt the access a single gateway, is one of the most significant challenges for future IoT networks. Due to a high latency during the transmissions of data packets caused by MAP, the deadlines for delay-critical applications can be missed. Much work has been conducted in recent years to address this problem.

This paper reviews the work on reactive and proactive approaches to MAP, showing that methods based on proactive (i.e., predictive) techniques are a highly promising avenue to mitigate MAP. The observations of the most recent research results can be recapitulated with the following remarks:

1. The diffusion analysis proposed in [19] for the probability of missed deadlines in MAP shows that the latency requirements of IoT devices can be met in networks with a massive number of devices by reducing the SCV of the interarrival times of packets.
2. The Randomization of Generation Times (RGT) preprocessing algorithm proposed by [36] significantly improves the performance of fast scheduling heuristics by randomizing generation time of packets with uniform distribution yielding an inter-arrival time SCV of close to 1/3.
3. The Quasi-Deterministic Transmission Policy (QDTP) [25] meets the deadlines of almost all packets for up to 6400 IoT devices by reducing the queue length at IoT gateways to nearly zero, and the SCV of packet inter-arival times is also brought down to nearly zero.

Acknowledgments. This research has been supported by the European Commission H2020 Program through the IoTAC Research and Innovation Action, under Grant Agreement No. 952684.

References

1. IoT Traffic Generation Pattern Dataset, January 2021. https://www.kaggle.com/tubitak1001118e277/iot-traffic-generation-patterns
2. Abdelrahman, O.H., Gelenbe, E.: A diffusion model for energy harvesting sensor nodes. In: 2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pp. 154–158. IEEE (2016)
3. Alavikia, Z., Ghasemi, A.: Collision-aware resource access scheme for LTE-based machine-to-machine communications. IEEE Trans. Veh. Technol. **67**(5), 4683–4688 (2018)
4. Ali, S., Rajatheva, N., Saad, W.: Fast uplink grant for machine type communications: challenges and opportunities. IEEE Commun. Mag. **57**(3), 97–103 (2019)
5. Astely, D., et al.: LTE release 14 outlook. IEEE Commun. Mag. **54**(6), 44–49 (2016)
6. Augusto-Gonzalez, J., et al.: From internet of threats to internet of things: a cyber security architecture for smart homes. In: 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 1–6. IEEE (2019)
7. Bello, O., Zeadally, S.: Toward efficient smartification of the internet of things (IoT) services. Future Gener. Comput. Syst. **92**, 663–673 (2019)
8. Chesnais, A., Gelenbe, E., Mitrani, I.: On the modeling of parallel access to shared data. Commun. ACM **26**(3), 196–202 (1983)
9. Cox, D.R., Miller, H.D.: The Theory of Stochastic Processes. Chapman and Hall, London (1965)
10. Du, J., Gelenbe, E., Jiang, C., Zhang, H., Ren, Y.: Contract design for traffic offloading and resource allocation in heterogeneous ultra-dense networks. IEEE J. Sel. Areas Commun. **35**(11), 2457–2467 (2017)
11. Eldeeb, E., Shehab, M., Alves, H.: A learning-based fast uplink grant for massive IoT via support vector machines and long short-term memory. IEEE Internet Things J. (2021)
12. Frötscher, A., Monschiebl, B., Drosou, A., Gelenbe, E., Reed, M.J., Al-Naday, M.: Improve cybersecurity of c-its road side infrastructure installations: the serIoT-secure and safe IoT approach. In: 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), pp. 1–5. IEEE (2019)
13. Gelenbe, E.: On approximate computer system models. J. ACM (JACM) **22**(2), 261–269 (1975)
14. Gelenbe, E., Caseau, Y.: The impact of information technology on energy consumption and carbon emissions. Ubiquity **2015**(June), 1–15 (2015)
15. Gelenbe, E., Ceran, E.T.: Energy packet networks with energy harvesting. IEEE Access **4**, 1321–1331 (2016). https://doi.org/10.1109/ACCESS.2016.2545340
16. Gelenbe, E., Mang, X., Feng, Y.: A diffusion cell loss estimate for ATM with multiclass bursty traffic. In: ATM 1995. IAICT, pp. 233–248. Springer, Boston (1996). https://doi.org/10.1007/978-0-387-35068-4_13

17. Gelenbe, E., Mitrani, I.: Analysis and Synthesis of Computer Systems, 2nd Edition. World Scientific Ltd. & Imperial College Press, London (2010). https://doi.org/10.1142/p643
18. Gelenbe, E., Nakip, M., Czachorski, T.: Improving massive access to an IoT gateway. Submitted for publication (2022)
19. Gelenbe, E., Nakip, M., Marek, D., Czachórski, T.: Diffusion analysis improves scalability of IoT networks to mitigate the massive access problem. In: 29th International Symposium on the Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS) (2021). (in Press)
20. Gelenbe, E., Ngai, E.: Adaptive random re-routing for differentiated QoS in sensor networks. Comput. J. **53**(7), 1052–1061 (2010)
21. Gelenbe, E., Ngai, E., Yadav, P.: Routing of high-priority packets in wireless sensor networks. In: IEEE Second International Conference on Computer and Network Technology, IEEE (2010)
22. Gelenbe, E., Ngai, E.C.H.: Adaptive QoS routing for significant events in wireless sensor networks. In: 2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, pp. 410–415. IEEE (2008)
23. Gelenbe, E., Pujolle, G.: Introduction to Networks of Queues. Wiley, Chichester (1998)
24. Gelenbe, E., Sevcik, K.: Analysis of update synchronization for multiple copy data bases. IEEE Trans. Comput. **10**, 737–747 (1979)
25. Gelenbe, E., Sigman, K.: IoT traffic shaping and the massive access problem. In: ICC 2022: IEEE International Conference on Communications, pp. 1–6. IEEE, May 2022
26. George, L., Rivierre, N., Spuri, M.: Preemptive and non-preemptive real-time uniprocessor scheduling (1996)
27. Ghavimi, F., Chen, H.H.: M2M communications in 3GPP LTE/LTE-A networks: architectures, service requirements, challenges, and applications. IEEE Commun. Surv. Tutorials **17**(2), 525–549 (2015)
28. Jang, H.S., Jin, H., Jung, B.C., Quek, T.Q.: Resource-optimized recursive access class barring for bursty traffic in cellular IoT networks. IEEE Internet Things J. (2021)
29. Jiang, N., Deng, Y., Nallanathan, A., Yuan, J.: A decoupled learning strategy for massive access optimization in cellular IoT networks. IEEE J. Sel. Areas Commun. **39**(3), 668–685 (2020)
30. Jin, H., Toor, W.T., Jung, B.C., Seo, J.B.: Recursive pseudo-Bayesian access class barring for M2M communications in LTE systems. IEEE Trans. Veh. Technol. **66**(9), 8595–8599 (2017)
31. Kim, H.-Y., Kim, J.-M.: A load balancing scheme based on deep-learning in IoT. Cluster Comput. **20**(1), 873–878 (2016). https://doi.org/10.1007/s10586-016-0667-5
32. Kleinrock, L.: Queueing Systems: Computer Applications. Wiley, Hoboken (1976)
33. Kobayashi, H.: Application of the diffusion approximation to queueing networks i: equilibrium queue distributions. J. ACM (JACM) **21**(2), 316–328 (1974)
34. Liang, L., Xu, L., Cao, B., Jia, Y.: A cluster-based congestion-mitigating access scheme for massive M2M communications in internet of things. IEEE Internet Things J. **5**(3), 2200–2211 (2018)
35. Liu, J., Song, L., et al.: A novel congestion reduction scheme for massive machine-to-machine communication. IEEE Access **5**, 18765–18777 (2017)

36. Nakip, M., Gelenbe, E.: Randomization of data generation times improves performance of predictive IoT networks. In: 2021 IEEE World Forum on Internet of Things (WF-IoT) (2021). (in Press)
37. Nakip, M., Gül, B.C., Rodoplu, V., Güzeliş, C.: Comparative study of forecasting schemes for IoT device traffic in machine-to-machine communication. In: Proceedings of the 2019 4th International Conference on Cloud Computing and Internet of Things, pp. 102–109 (2019)
38. Nakip, M., Karakayali, K., Güzeliş, C., Rodoplu, V.: An end-to-end trainable feature selection-forecasting architecture targeted at the internet of things. IEEE Access **9**, 1–1 (2021). https://doi.org/10.1109/ACCESS.2021.3092228
39. Nakip, M., Rodoplu, V., Güzeliş, C., Eliiyi, D.T.: Joint forecasting-scheduling for the internet of things. In: 2019 IEEE Global Conference on Internet of Things (GCIoT), pp. 1–7. IEEE (2019)
40. Natsiavas, P., et al.: Developing an infrastructure for secure patient summary exchange in the EU context: lessons learned from the konfido project. Health Inf. J. **27**(2), 14604582211021460 (2021)
41. Newell, G.F.: Applications of Queueing Theory. Chapman and Hall, London, June 1971
42. Ngai, E.C.H., Gelenbe, E., Humber, G.: Information-aware traffic reduction for wireless sensor networks. In: 2009 IEEE 34th Conference on Local Computer Networks, pp. 451–458. IEEE (2009)
43. Petkov, V., Obraczka, K.: The case for using traffic forecasting in schedule-based channel access. In: 2011 IEEE Consumer Communications and Networking Conference (CCNC), pp. 208–212. IEEE (2011)
44. Petkov, V., Obraczka, K.: Collision-free medium access based on traffic forecasting. In: 2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1–9. IEEE (2012)
45. Raca, D., et al.: On leveraging machine and deep learning for throughput prediction in cellular networks: design, performance, and challenges. IEEE Commun. Mag. **58**(3), 11–17 (2020)
46. Rodoplu, V., Nakıp, M., Eliiyi, D.T., Güzelis, C.: A multi-scale algorithm for joint forecasting-scheduling to solve the massive access problem of IoT. IEEE Internet Things J. (2020)
47. Rodoplu, V., Nakip, M., Qorbanian, R., Eliiyi, D.T.: Multi-channel joint forecasting-scheduling for the internet of things. IEEE Access **8**, 217324–217354 (2020)
48. Ruan, L., Dias, M.P.I., Wong, E.: Machine learning-based bandwidth prediction for low-latency H2M applications. IEEE Internet Things J. **6**(2), 3743–3752 (2019)
49. Shahin, N., Ali, R., Kim, Y.T.: Hybrid slotted-CSMA/CA-TDMA for efficient massive registration of IoT devices. IEEE Access **6**, 18366–18382 (2018)
50. Shehab, M., Hagelskjær, A.K., Kalør, A.E., Popovski, P., Alves, H.: Traffic prediction based fast uplink grant for massive IoT. In: 2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1–6. IEEE (2020)
51. Shirvanimoghaddam, M., Dohler, M., Johnson, S.J.: Massive non-orthogonal multiple access for cellular IoT: potentials and limitations. IEEE Commun. Mag. **55**(9), 55–61 (2017)
52. Soltanmohammadi, E., Ghavami, K., Naraghi-Pour, M.: A survey of traffic issues in machine-to-machine communications over LTE. IEEE Internet Things J. **3**(6), 865–884 (2016)

53. Takács, L.: Introduction to the Theory of Queues. Oxford University Press, Oxford (1962)
54. Tarab, H.: Real time performance testing of LoRa-LPWAN based environmental monitoring UAV system. University of Windsor, Electronic Theses and Dissertations. 7578 (2018). https://scholar.uwindsor.ca/etd/7578
55. Tello-Oquendo, L., et al.: Performance analysis and optimal access class barring parameter configuration in LTE-A networks with massive M2M traffic. IEEE Trans. Veh. Technol. **67**(4), 3505–3520 (2018)
56. Tello-Oquendo, L., Pacheco-Paramo, D., Pla, V., Martinez-Bauset, J.: Reinforcement learning-based ACB in LTE-A networks for handling massive M2M and H2H communications. In: 2018 IEEE International Conference on Communications (ICC), pp. 1–7. IEEE (2018)

Author Index