6. Conclusion

In this last chapter, Rosie is discussed. Some of its advantages, and disadvantages are considered. Furthermore, some next steps for future work are proposed.

6.1. System Discussion

Even though there is extensive research behind smart environments and activity identification, there is limited research regarding securing the home. Therefore, taking the extensive research on another discipline as a starting point is beneficial. Nevertheless, a sizable disadvantage is the considerable amount of research produced by a discipline that is over 60 years old. In consequence, identifying the appropriate applicable techniques to be used, is quite time consuming and resource intensive.

Furthermore, current research on machine learning algorithms for computer networks rely heavily on standardized training sets. Using the same training sets for comparisons, establishes a baseline against which different machine learning algorithms can be compared. Nevertheless, on their findings many researchers suggest that the datasets needed to be cleaned up, or also processed to extract more features. Unfortunately, not included in the findings are the list of new features, or the resulting dataset after preprocessing. In the context of this research, this is a problem because the type of features that can be extracted from the data provide a lot of information on how their machine learning algorithms work.

From the smart home point of view, the problem is the opposite. There are no standardized or agreed upon training sets for smart home that are publicly available. In my opinion, the root cause of this problem is that technologies and protocols for computer networks are much more standardized than those for smart environments. Second, the topology of computer networks has also been studied much more than sensor topologies for the smart home environment. This results in an easier job at standardization of datasets for the networking context.

From the machine learning point of view, there are some disadvantages from the Rosie system. First, the approach taken by Selim *et al.* [45] while developing their hybrid network intrusion detection algorithm is very interesting. They developed a layered architecture for each aspect of the detection process, then implemented seven machine learning algorithms and then tested each one of them with the data for each layer. After testing, they did a complete statistical analysis to decide which algorithm was more suitable for each layer of the architecture. This type of approach to selecting algorithms may be beneficial to Rosie; Testing a wider range of algorithms, and statistically analyzing which ones and under what conditions perform better. Another disadvantage of the system, is how the algorithms learn. On one hand, it is not possible to have bulk data to or off-line training. Each user in every house behaves different. On the other hand, it is not possible to train a machine learning algorithm without data. Therefore, there will be a time from the point when the system is started, until the point the algorithm is trained, that no predictions can be achieved. Using on-line learning algorithms may be a way to tackle that problem. However, the compromise is that this approach requires more user, when interacting with notifications about currently sensed events. The policy used by Rosie to counter this problem, is to let the algorithms train for the first week, then predict using the trained model for another week, while at the same time training a second model. After the end of the week, the old model is discarded and swapped for the recently trained one.

During the development of the application domain models, the selected methodology dictates that the analysis objects must be classified into boundary, entity, and control objects. Nevertheless, In the case of these research, the classification of the fixtures was not straightforward. They can be classified as entity objects because their data is the one tracked by the system. However, they also could be classified as boundary objects, because they represent one of the system interfaces with the houseÂ´s inhabitants. Users of the system interact directly with the fixtures, even if they do not realize it. The conclusion reached, was to have two types of objects, one representing the data, and the other one describing the interaction element.

From the architectural point of view, by implementing the blackboard pattern, the system is able to use different knowledge sources that only depend on the blackboard association, and how the tuples are stored and read from the blackboard. This implies that knowledge sources function as independent, self-contained pieces of code. On one hand, this helps localization, and responsibility assignment, but on the other hand allows to add easily new knowledge sources.

Even though the current implementation of Rosie does not implement it, It is possible to create new knowledge sources that communicate with external machine learning systems. For example, it is conceivable to write an expert that takes the data stored on the blackboard and translates it into Python commands. That way deep learning libraries written in other languages, that take advantages of GPUs, can be also used. It is also possible to use web-based machine learning frameworks, to off-load the burden of running those algorithms from what may be the underpowered device that runs Rosie.

By selecting OpenHAB as the execution environment, a couple of compromises must be made. First, when setting up a sensor, OpenHAB allows to define information such as types, and also extra data such as sensor groups. Nevertheless, not all of that information is accessible to the bindings. When the value for one of the items is read from the event bus, the only information available is the name of the item and the new value. This seems like an unnecessary impairment that handicaps the functionality of the bindings. Specially, considering that such meta data may be useful, in particular for automation applications. It is important to note that the meta data is not lost, it is simply not accessible from the event bus. Yet, other interfaces have access to it. However, there are a few of workarounds. The bindings can directly query OpenHAB's own REST interface to access that meta data. However, accessing that meta data through a web interface seems cumbersome and error-prone.

Another workaround, the one used by Rosie, is to use the configuration mechanism for the bindings, and write the meta data there as well to be later stored on the binding. The downside from this approach, is that the configuration files have now duplicated information. When the amount of sensors is large, maintaining the configuration files becomes cumbersome.

Nevertheless, the advantages of using OpenHAB outweigh this drawback. Having the integration of different home automation systems and technologies, into one single solution, provides Rosie with out-of-the-box extensibility. Furthermore, its message bus architectural style and OSGi container, enable the implementation of additional components, like the simulator used for testing, that mimic the events of a real home.

Another architectural aspect, is the classification of sensors. Dividing the sensors into motion, presence, environmental, electrical, and point of entry, allow for the experts to request them from to blackboard according to type. That way, the knowledge sources can easily and directly ask for the information that they require. For example, an expert may only query the information from the presence data and the motion data, to correlate it without needing to explicitly know all the names of all the available motion of presence sensors.

Furthermore, this classification allows RosieÂ´s experts to be resilient to data loss or sensor failure. When a sensors comes off-line, the hypotheses may not be reached, but the overall functionality of the system will continue to operate.

Having this classification, also allows for more specific data experts to be written. An expert could be developed with some basic rules that characterize normal interactions between the sensors. Not to be used for detection of anomalies, but to be used as classifiers that can pre-classify data to later be fed to the machine learning algorithms. This would reduce the amount of interaction that the user has with the system for on-line machine learning algorithms.

Furthermore, the separation of responsibilities from the current architecture, allows new data extractions experts to be written, that work at the same time with the existing ones. Enabling exploration of new contexts to find different, or even better, features to be extracted.

In my opinion, the advantages turn Rosie into a platform for developing, deploying, and testing machine learning algorithms on smart environments.

6.2. Future Work

Both the advantages and disadvantages described in the last section provide opportunities to build upon the work presented in this thesis, and expand the research on the smart home environments. From this perspective, one of the identified opportunities is to build better simulations tools. It is difficult for universities to model the behavior of occupants of a home, without building houses and asking researchers to move into them. By having simulations tools, that also connect easily to frameworks like OpenHAB, research will benefit from being able to test multiple algorithms, in multiple models of homes, with different types of occupants and lifestyles, without the need to build multiple houses. Furthermore, with simulation software it is possible to make the passage of time go faster. Training of machine learning algorithms would go rapidly, to later be tested even on real-world scenarios.

The advantage in developing Rosie is the ease of use when testing automation algorithms on the smart home environment. Using this advantage, developing new machine learning algorithms for the context of smart environments would make development more swift. From the machine learning perspective, it would be very interesting to develop a new set of features, maybe even based on another discipline, and using Rosie as a test platform for analysis.

Also interesting would be connecting Rosie to another web-based system. First, having the power of a data center behind Rosie enables the prospect of using more powerful but also more resource-intensive algorithms. Furthermore, the question of what would happen if Rosie is installed in more than one home is also interesting. If all the houses on a block are equipped with Rosie, they could coordinate the threat level between them, making not only one home safer, but the entire neighborhood.

Appendix

A. Features Extracted From Networking Datasets

Table A.1.: Packet level features of the TUIDS intrusion dataset (Part 1)

Feature names	Feature description
Basic features	
Duration	Time since occurrence of the first frame
Protocol	Protocol of layer 3- IP, TCP, UDP
Src IP	Source IP address
Dst IP	Destination IP address
Src port	Source port of machine
Dst port	Destination port of machine
Service	Network service on the destination e.g. http, telnet, etc.
num-bytes-src-dst	No. of data bytes flowing from src to dst
num-bytes-dst-src	No. of data bytes flowing from dst to src
Fr-no.	Frame number
Fr-length	Length of the frame
Cap-length	Captured frame length
Head-len	Header length of the packet
Frag-offset	Fragment offset value
TTL	Time to live
Seq-no.	Sequence number
CWR	Congestion window record
ECN	Explicit congestion notification
URG	Urgent TCP flag
ACK	Ack flag
PSH	Push TCP flag
RST	Reset RST flag
SYN	Syn TCP flag
FIN	Fin TCP flag

Table A.2.: Packet level features of the TUIDS intrusion dataset (Part 2)

Feature names	Feature description
Content-based features	
Land	1 if connection is from/to the same host/port; 0 otherwise
Mss-src-dst-requested	Maximum segment size from src to dst requested
Mss-dst-src-requested	Maximum segment size from dst to src requested
Ttt-len-src-dst	Time to live length from src to dst
Ttt-len-dst-src	Time to live length from dst to src
Conn-status	Status of the connection (1-complete, 0-reset)
Time-based features	
count-fr-dst	No. of frames received by the unique dst in the last T sec from the same src
count-fr-src	No. of frames received by the unique src in the last T sec to the same dst
count-serv-src	No. of frames from the src to the same dst port in the last T sec
count-serv-dst	No. of frames from dst to the same src port in the last T sec
num-pushed-src-dst	No. of pushed pkts flowing from src to dst
num-pushed-dst-src	No. of pushed pkts flowing from dst to src
num-SYN-FIN-src-dst	No. of SYN/FIN pkts flowing from src to dst
num-SYN-FIN-dst-src	No. of SYN/FIN pkts flowing from dst to src
num-FIN-src-dst	No. of FIN pkts flowing from src to dst
num-FIN-dst-src	No. of FIN pkts flowing from dst to src
Connection-based features	
count-dst-conn	No. of frames to the unique dst in the last N packets from the same src
count-src-conn	No. of frames from the unique src in the last N packets to the same dst
count-serv-src-conn	No. of frames from the src to the same dst port in the last N packets
count-serv-dst-conn	No. of frames from the dst to the same src port in the last N packets
num-packets-src-dst	No. of packets flowing from src to dst
num-packets-dst-src	No. of packets flowing from dst to src
num-acks-src-dst	No. of ack packets flowing from src to dst
num-acks-dst-src	No. of ack packets flowing from dst to src
num-retransmit-src-dst	No. of retransmitted packets flowing from src to dst
num-retransmit-dst-src	No. of retransmitted packets flowing from dst to src

Table A.3.: Flow level features of the TUIDS intrusion dataset

Feature names	Feature description
Basic features	
Duration	Length of the flow (in seconds)
Protocol-type	Type of protocols - TCP, UDP, ICMP
Src IP	Src node IP address
Dst IP	Destination IP address
Src port	Source port
Dst port	Destination port
ToS	Type of service
URG	Urgent flag of TCP header
ACK	Ack flag
PSH	Push flag
RST	Reset flag
SYN	SYN flag
FIN	FIN flag
Source byte	No. of data bytes transferred from the src IP addrs to the dst IP addrs
Dst byte	No. of data bytes transferred from the dst IP addrs to the src IP addrs
Land	1 if connection is from/to the same host/port; 0 otherwise
Time-window features	
Count-dst	No. of flows to the unique dst IP addr inside the network in the last T sec from the same src
Count-src	No. of flows from the unique src IP addr inside the network in the last T sec to the same dst
Count-serv-src	No. of flows from the src IP to the same dst port in the last T sec
Count-serv-dst	No. of flows to the dst IP using the same src port in the last T sec
Connection-based features	
Count-dst-conn	No. of flows to the unique dst IP addrs in the last N flows from the same src
Count-src-conn	No. of flows from the unique src IP addrs in the last N flows to the same dst
Count-serv-src-conn	No. of flows from the src IP addrs to the same dst port in the last N flows.
Count-serv-dst-conn	No. of flows to the dst IP addrs to the same src port in the last N flows

Table A.4.: Features of the DDoS dataset

Feature names	**Feature description**
Duration	Length of the flow (in seconds)
Protocol-type	Type of protocols - TCP, UDP, ICMP
src IP	Source node IP address
dest IP	Destination IP address
src port	Source port
dest port	Destination port
ToS	Type of service
URG	Urgent flag of TCP header
ACK	Ack flag
PSH	Push flag
RST	Reset flag
SYN	SYN flag
FIN	FIN flag
src byte	No. of data bytes transferred from the src IP addrs to the dst IP addrs
Land	1 if connection is from/to the same host/port; 0 otherwise
count-dst	No. of flows to the unique dst IP addrs inside the network in the last T sec (5 s) from the same src
count-src	No. of flows from the unique src IP addrs inside the network in the last T sec (5 s) to the same dst
count-serv-src	No. of flows from the src IP addrs to the same dst port in the last T sec (5 s)
count-serv-dst	No. of flows to the dst IP addrs using the same src port in the last T sec (5 s)
count-dst-conn	No. of flows to the unique dst IP addrs in the last N flows from the same src
count-src-conn	No. of flows from the unique src IP addrs in the last N flows to the same dst
count-serv-src-conn	No. of flows from the src IP addrs to the same dst port in the last N flows
count-serv-dst-conn	No. of flows to the dst IP addrs to the same src port in the last N flows
Label	Label of the feature instance as normal/attack

Bibliography

[1] M. Bahrololum and M. Khaleghi. "Anomaly Intrusion Detection System Using Gaussian Mixture Model". In: *2008 Third International Conference on Convergence and Hybrid Information Technology*. Vol. 1. IEEE, Nov. 2008, pp. 1162–1167. ISBN: 978-0-7695-3407-7. DOI: 10.1109/ICCIT.2008.17. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4682192.

[2] B Balajinath and S.V Raghavan. "Intrusion Detection Through Learning Behavior Model". In: *Comput. Commun.* 24.12 (July 2001), pp. 1202–1212. ISSN: 0140-3664. DOI: 10.1016/S0140-3664(00)00364-9. URL: http://dx.doi.org.eaccess.ub.tum.de/10.1016/S0140-3664(00)00364-9.

[3] Daniel Barbará et al. "ADAM: A Testbed for Exploring the Use of Data Mining in Intrusion Detection". In: *SIGMOD Rec.* 30.4 (Dec. 2001), pp. 15–24. ISSN: 0163-5808. DOI: 10.1145/604264.604268. URL: http://doi.acm.org.eaccess.ub.tum.de/10.1145/604264.604268.

[4] S. Jennifer Beaudin, S. Stephen Intille, and E. Margaret Morris. "To Track or Not to Track: User Reactions to Concepts in Longitudinal Health Monitoring". In: *J Med Internet Res* 8.4 (Dec. 7, 2006), e29. DOI: 10.2196/jmir.8.4.e29. URL: http://www.jmir.org/2006/4/e29/.

[5] Dhruba Kumar Bhattacharyya and Jugal Kumar Kalita. *Network Anomaly Detection: A Machine Learning Perspective*. Chapman & Hall/CRC, 2013. ISBN: 1466582081, 9781466582088.

[6] Erwin A. Blackstone, Andrew J. Buck, and Simon Hakim. "Evaluation of alternative policies to combat false emergency calls". In: *Evaluation and Program Planning* 28.2 (2005), pp. 233–242. ISSN: 0149-7189. DOI: http://dx.doi.org/10.1016/j.evalprogplan.2004.09.004. URL: http://www.sciencedirect.com/science/article/pii/S0149718905000121.

[7] Rich Brown. *A washer in your washer and smart-home sprawl at CES 2015*. CNET. Apr. 2015. URL: http://www.cnet.com/news/a-washer-in-your-washer-and-smart-home-sprawl-at-ces-2015/.

[8] Bernd Brügge and Allen H. Dutoit. *Object-oriented software engineering - using UML, patterns and Java (2nd ed.)* Prentice Hall, 2004. ISBN: 978-0-13-191179-6.

[9] Frank Buschmann et al. *Pattern-oriented Software Architecture: A System of Patterns*. New York, NY, USA: John Wiley & Sons, Inc., 1996. ISBN: 0-471-95869-7.

[10] Pedro Casas, Johan Mazel, and Philippe Owezarski. "UNADA: Unsupervised Network Anomaly Detection Using Sub-space Outliers Ranking". English. In: *NETWORKING 2011*. Ed. by Jordi Domingo-Pascual et al. Vol. 6640. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, pp. 40–51. ISBN: 978-3-642-20756-3. DOI: 10.1007/978-3-642-20757-0_4. URL: http://dx.doi.org/10.1007/978-3-642-20757-0_4.

[11] Rung-Ching Chen et al. "Using Rough Set and Support Vector Machine for Network Intrusion Detection System". In: *Intelligent Information and Database Systems, 2009. ACIIDS 2009. First Asian Conference on*. Apr. 2009, pp. 465–470. DOI: 10.1109/ACIIDS.2009.59.

[12] P. Chhabra et al. "Distributed Spatial Anomaly Detection". In: *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. Apr. 2008. DOI: 10.1109/INFOCOM.2008.232.

[13] D.J. Cook et al. "MavHome: an agent-based smart home". In: *Pervasive Computing and Communications, 2003. (PerCom 2003). Proceedings of the First IEEE International Conference on*. Mar. 2003, pp. 521–524. DOI: 10.1109/PERCOM.2003.1192783.

[14] SajalK. Das and DianeJ. Cook. "Designing Smart Environments: A Paradigm Based on Learning and Prediction". English. In: *Pattern Recognition and Machine Intelligence*. Ed. by SankarK. Pal, Sanghamitra Bandyopadhyay, and Sambhunath Biswas. Vol. 3776. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 80–90. ISBN: 978-3-540-30506-4. DOI: 10.1007/11590316_11. URL: http://dx.doi.org/10.1007/11590316_11.

[15] J.E. Dickerson and J.A. Dickerson. "Fuzzy network profiling for intrusion detection". In: *Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American*. 2000, pp. 301–306. DOI: 10.1109/NAFIPS.2000.877441.

[16] Lee D. Erman et al. "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty". In: *ACM Comput. Surv.* 12.2 (June 1980), pp. 213–253. ISSN: 0360-0300. DOI: 10.1145/356810.356816. URL: http://doi.acm.org.eaccess.ub.tum.de/10.1145/356810.356816.

[17] C. Fung and R. Boutaba. "Intrusion Detection Networks: A Key to Distributed Security". In: CRC Press, 2013. ISBN: 978-1-4665-6412-1.

[18] Erich Gamma et al. *Design Patterns: Elements of Reusable Object-oriented Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995. ISBN: 0-201-63361-2.

[19] Zoubin Ghahramani. "Hidden Markov Models". In: River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2002. Chap. An Introduction to Hidden Markov Models and Bayesian Networks, pp. 9–42. ISBN: 981-02-4564-5. URL: `http://dl.acm.org/citation.cfm?id=505741.505743`.

[20] Prasanta Gogoi, Bhogeswar Borah, and Dhruba K Bhattacharyya. "Network Anomaly Detection Using Unsupervised Model". In: *IJCA Special Issue on Network Security and Cryptography* NSC.1 (Dec. 2011). Full text available, pp. 19–30.

[21] Prasanta Gogoi et al. "A Survey of Outlier Detection Methods in Network Anomaly Identification". In: *Comput. J.* 54.4 (Apr. 2011), pp. 570–588. ISSN: 0010-4620. DOI: `10.1093/comjnl/bxr026`. URL: `http://dx.doi.org/10.1093/comjnl/bxr026`.

[22] Prasanta Gogoi et al. "MLH-IDS: A Multi-Level Hybrid Intrusion Detection Method". In: *The Computer Journal* 57.4 (2014), pp. 602–623. DOI: `10.1093/comjnl/bxt044`. eprint: `http://comjnl.oxfordjournals.org/content/57/4/602.full.pdf+html`. URL: `http://comjnl.oxfordjournals.org/content/57/4/602.abstract`.

[23] Mark Hall et al. "The WEKA Data Mining Software: An Update". In: *SIGKDD Explor. Newsl.* 11.1 (Nov. 2009), pp. 10–18. ISSN: 1931-0145. DOI: `10.1145/1656274.1656278`. URL: `http://doi.acm.org/10.1145/1656274.1656278`.

[24] IFTTT Inc. *ifttt the beginning*. IFTTT Inc. 111 2015. URL: `http://blog.ifttt.com/`.

[25] IFTTT Inc. *What is IFTTT?* IFTTT Inc. Nov. 2015. URL: `https://ifttt.com/wtf`.

[26] Stephen S. Intille. *The Goal: Smart People, Not Smart Homes*. URL: `http://web.media.mit.edu/~intille/papers-files/IntilleICOST06.pdf`.

[27] StephenS. Intille et al. “Using a Live-In Laboratory for Ubiquitous Computing Research”. English. In: *Pervasive Computing*. Ed. by KennethP. Fishkin et al. Vol. 3968. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2006, pp. 349–365. ISBN: 978-3-540-33894-9. DOI: 10.1007/11748625_22. URL: http://dx.doi.org/10.1007/11748625_22.

[28] S.S. Intille K.C. Cheung and K. Larson. “An Inexpensive Bluetooth-Based Indoor Positioning Hack”. In: *Proceedings of UbiComp 2006 Extended Abstracts (Posters Program)*. 2006. URL: http://web.mit.edu/cron/group/house_n/documents/CheungIntilleLarson2006.pdf.

[29] CoryD. Kidd et al. “The Aware Home: A Living Laboratory for Ubiquitous Computing Research”. English. In: *Cooperative Buildings. Integrating Information, Organizations, and Architecture*. Ed. by NorbertA. Streitz et al. Vol. 1670. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1999, pp. 191–198. ISBN: 978-3-540-66596-0. DOI: 10.1007/10705432_17. URL: http://dx.doi.org/10.1007/10705432_17.

[30] Julie A. Kientz et al. “Grow and Know: Understanding Record-keeping Needs for Tracking the Development of Young Children”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI ’07. San Jose, California, USA: ACM, 2007, pp. 1351–1360. ISBN: 978-1-59593-593-9. DOI: 10.1145/1240624.1240830. URL: http://doi.acm.org/10.1145/1240624.1240830.

[31] Julie A. Kientz et al. “The Georgia Tech Aware Home”. In: *CHI ’08 Extended Abstracts on Human Factors in Computing Systems*. CHI EA ’08. Florence, Italy: ACM, 2008, pp. 3675–3680. ISBN: 978-1-60558-012-8. DOI: 10.1145/1358628.1358911. URL: http://doi.acm.org/10.1145/1358628.1358911.

[32] Anna Koufakou and Michael Georgiopoulos. “A Fast Outlier Detection Strategy for Distributed High-dimensional Data Sets with Mixed Attributes”. In: *Data Min. Knowl. Discov.* 20.2 (Mar. 2010), pp. 259–289. ISSN: 1384-5810. DOI: 10.1007/s10618-009-0148-z. URL: http://dx.doi.org/10.1007/s10618-009-0148-z.

[33] Christopher Kruegel et al. “Bayesian Event Classification for Intrusion Detection”. In: *Proceedings of the 19th Annual Computer Security Applications Conference*. ACSAC ’03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 14–. ISBN: 0-7695-2041-3. URL: http://dl.acm.org/citation.cfm?id=956415.956436.

[34] Pat Langley, Wayne Iba and, and Kevin Thompson. "An Analysis of Bayesian Classifiers". In: *Proceedings of the Tenth National Conference on Artificial Intelligence*. AAAI'92. San Jose, California: AAAI Press, 1992, pp. 223–228. ISBN: 0-262-51063-4. URL: `http://dl.acm.org/citation.cfm?id=1867135.1867170`.

[35] S.C. Lee and D.V. Heinbuch. "Training a neural-network based intrusion detector to recognize novel attacks". In: *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on* 31.4 (July 2001), pp. 294–299. ISSN: 1083-4427. DOI: `10.1109/3468.935046`.

[36] S. Mabu et al. "An Intrusion-Detection Model Based on Fuzzy Class-Association-Rule Mining Using Genetic Network Programming". In: *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on* 41.1 (Jan. 2011), pp. 130–139. ISSN: 1094-6977. DOI: `10.1109/TSMCC.2010.2050685`.

[37] Tara Matthews et al. "A Toolkit for Managing User Attention in Peripheral Displays". In: *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*. UIST '04. Santa Fe, NM, USA: ACM, 2004, pp. 247–256. ISBN: 1-58113-957-8. DOI: `10.1145/1029632.1029676`. URL: `http://doi.acm.org.eaccess.ub.tum.de/10.1145/1029632.1029676`.

[38] Michael Mozer. "Lessons from an adaptive house". PhD thesis. Architectural Engineering, 2004.

[39] Inc. Open Source Robotics Foundation. *Documentation*. Open Source Robotics Foundation, Inc. Nov. 2015. URL: `http://wiki.ros.org/`.

[40] Inc. Open Source Robotics Foundation. *History*. Open Source Robotics Foundation, Inc. Nov. 2015. URL: `http://www.ros.org/history/`.

[41] Zdzislaw Pawlak. "Rough set approach to knowledge-based decision support". In: *European Journal of Operational Research* 99.1 (1997), pp. 48–57. ISSN: 0377-2217. DOI: `http://dx.doi.org/10.1016/S0377-2217(96)00382-7`. URL: `http://www.sciencedirect.com/science/article/pii/S0377221796003827`.

[42] Dhanji R. Prasanna. *Dependency Injection*. 1st. Greenwich, CT, USA: Manning Publications Co., 2009. ISBN: 193398855X, 9781933988559.

[43] Parisa Rashidi et al. "Inhabitant Guidance of Smart Environments". English. In: *Human-Computer Interaction. Interaction Platforms and Techniques*. Ed. by JulieA. Jacko. Vol. 4551. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, pp. 910–919. ISBN: 978-3-540-73106-1. DOI: `10.1007/978-`

3-540-73107-8_100. URL: http://dx.doi.org/10.1007/978-3-540-73107-8_100.

[44] K. Larson S. S. Intille and E. M. Tapia. "Designing and evaluating technology for independent aging in the home". In: *Proceedings of the International Conference on Aging, Disability and Independence*. 2003. URL: http://web.media.mit.edu/~intille/papers-files/IntilleLarsonTapia03.pdf.

[45] Mohamed Hashem Sahar Selim and Taymoor M. Nazmy. "Hybrid Multi-level Intrusion Detection System". In: *International Journal of Computer Science and Information Security*. Vol. 9. 5. 2011, pp. 23–29.

[46] *Smart Home*. Google trends. Apr. 2015. URL: https://www.google.com/trends/explore%5C#q=smart%20home.

[47] Thomas Stibor, Jonathan Timmis, and Claudia Eckert. "A Comparative Study of Real-Valued Negative Selection to Statistical Anomaly Detection Techniques". English. In: *Artificial Immune Systems*. Ed. by Christian Jacob et al. Vol. 3627. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pp. 262–275. ISBN: 978-3-540-28175-7. DOI: 10.1007/11536444_20. URL: http://dx.doi.org/10.1007/11536444_20.

[48] Arman Tajbakhsh, Mohammad Rahmati, and Abdolreza Mirzaei. "Intrusion detection using fuzzy association rules". In: *Applied Soft Computing* 9.2 (2009), pp. 462–469. ISSN: 1568-4946. DOI: http://dx.doi.org/10.1016/j.asoc.2008.06.001. URL: http://www.sciencedirect.com/science/article/pii/S1568494608000975.

[49] OpenHAB UG. *How to Implement a Binding*. OpenHAB UG. July 2015. URL: https://github.com/openhab/openhab/wiki/How-To-Implement-A-Binding.

[50] OpenHAB UG. *Introduction*. OpenHAB UG. July 2015. URL: http://www.openhab.org/features/introduction.html.

[51] OpenHAB UG. *OpenHAB Items*. OpenHAB UG. July 2015. URL: https://github.com/openhab/openhab/wiki%5C#openhab-runtime.

[52] OpenHAB UG. *OpenHAB Runtime*. OpenHAB UG. July 2015. URL: https://github.com/openhab/openhab/wiki%5C#openhab-runtime.

[53] OpenHAB UG. *Supported platforms*. OpenHAB UG. July 2015. URL: http://www.openhab.org/features/supported-platforms.html.

[54] University of Waikato. *Class IBk*. University of Waikato. July 2015. URL: https://github.com/openhab/openhab/wiki%5C#openhab-runtime.

[55] Darrell Whitley. "A genetic algorithm tutorial". English. In: *Statistics and Computing* 4.2 (1994), pp. 65–85. ISSN: 0960-3174. DOI: 10.1007/BF00175354. URL: http://dx.doi.org/10.1007/BF00175354.

[56] Megan Wollerton. *The smart home is a hit at IFA 2015*. CNET. Nov. 2015. URL: http://www.cnet.com/pictures/the-smart-home-is-a-hit-at-ifa-2015-pictures/.