References

[1] W. Stallings, *Computer organization and architecture: designing for performance*. Upper Saddle Rive, N.J.: Pearson Prentice Hall, 2006.

[2] J. Fuegi and Jo Francis, “Lovelace & Babbage and the Creation of the 1843 ‘Notes,’” *IEEE Ann. Hist. Comput.*, vol. 25, no. 4, pp. 16–26, Dec. 2003, doi: 10.1109/MAHC.2003.1253887.

[3] D. A. Patterson and J. L. Hennessy, *Computer organization and design*, 5th ed. oxford: Morgan Kaufmann Publishers, 2014.

[4] R. Das, “Blurring the Lines between Memory and Computation,” *IEEE Micro*, vol. 37, no. 6, pp. 13–15, Nov. 2017, doi: 10.1109/MM.2017.4241340.

[5] S. A. McKee and others, “Reflections on the memory wall.,” in *Conf. Computing Frontiers*, 2004, p. 162.

[6] P. Machanick, “Approaches to addressing the memory wall,” *Sch. IT Electr. Eng. Univ. Qld.*, 2002.

[7] D. A. Patterson and J. L. Hennessy, *Computer organization and design*, RISC-V. Morgan Kaufmann Publishers, 2018.

[8] A. Jog *et al.*, “Anatomy of GPU Memory System for Multi-Application Execution,” in *Proceedings of the 2015 International Symposium on Memory Systems*, New York, NY, USA, 2015, pp. 223–234, doi: 10.1145/2818950.2818979.

[9] K. Wang, X. Ding, R. Lee, S. Kato, and X. Zhang, “GDM: device memory management for gpgpu computing,” *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 1, pp. 533–545, 2014.

[10] G. W. Burr *et al.*, “Phase change memory technology,” *J. Vac. Sci. Technol. B Nanotechnol. Microelectron. Mater. Process. Meas. Phenom.*, vol. 28, no. 2, pp. 223–262, 2010.

[11] C. Neumann, S. S. Qin, and X. Zheng, “Stanford Memory Trends,” *Stanford Nanoelectronics Lab*, 2018. https://nano.stanford.edu/stanford-memory-trends (accessed Mar. 29, 2019).

[12] S. Ambareesh and J. Fathima, “A Cached Middleware to Improve I/O Performance Using POSIX Interface,” *Procedia Comput. Sci.*, vol. 85, pp. 125–132, 2016, doi: 10.1016/j.procs.2016.05.197.

[13] M. Abd-El-Barr and H. El-Rewini, *Fundamentals of computer organization and architecture*. Hoboken, N.J: Wiley, 2005.

[14] G. E. Moore, “Moore’s Law at 40. Chapter 7,” in *Understanding Moore’s Law: Four decades of innovation edited by D. C. Brock*, Philadelphia, PA.: Chemical Heritage Foundation, 2006, pp. 67–84.

[15] A. S. Karunananda, P. R. Goldin, G. Rzevski, S. Fernando, and H. R. Fernando, “ON COMPUTING THE BEHAVIOR OF THE MIND FROM AN EASTERN PHILOSOPHICAL PERSPECTIVE,” *Int. J. Des. Nat. Ecodynamics*, vol. 10, no. 3, pp. 224–232, 2015, doi: 10.2495/DNE-V10-N3-224-232.

[16] W. A. C. Weerakoon, A. S. Karunananda, and N. G. J. Dias, "Six-state Continuous Processing Model for a New Theory of Computing," University of Moratuwa, 2019, vol. 890, pp. 32–48, doi: https://doi.org/10.1007/978-981-13-9129-3_3.

[17] W. A. C. Weerakoon, A. S. Karunananda, and N. G. J. Dias, "A tactics memory for a new theory of computing," Apr. 2013, pp. 153–158, doi: 10.1109/ICCSE.2013.6553901.

[18] W. A. C. Weerakoon, A. S. Karunananda, and N. G. J. Dias, "Conditionally evolving memory for computers," Aug. 2015, pp. 271–271, doi: 10.1109/ICTER.2015.7377704.

[19] A. S. Karunananda, "Computer modelling of thought processes," *Int. J. Comput. Appl. Technol.*, vol. 6, no. 2/3, pp. 135–140, 1993.

[20] Karunananda A. S., "Using an eastern philosophy for providing a theoretical basis for some Heuristics used in artificial neural networks," *Malayasian J. Comput. Sci.*, vol. 15, no. 2, pp. 28–33, Dec. 2002.

[21] L. P. Ranatunga, "ON COMPUTING MENTAL STATES," *Malays. J. Comput. Sci.*, vol. 9, no. 2, pp. 43–53, 1996.

[22] W. Stallings, *Operating systems: internals and design principles*, 7th ed. Boston: Prentice Hall, 2012.

[23] A. S. Tanenbaum and H. Boss, *Modern operating systems*, 4th ed. Upper Saddle River, N.J: Pearson/Prentice Hall, 2015.

[24] H. Kasim, V. March, R. Zhang, and S. See, "Survey on parallel programming model," in *InIFIP International Conference on Network and Parallel Computing*, Oct. 2008, pp. 266–275.

[25] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. 2015.

[26] M. J. Wooldridge, *An introduction to multiagent systems*, 2nd ed. Chichester, U.K: John Wiley & Sons, 2009.

[27] Black Paul E., D. Richard Kuhn, and Carl J. Williams, "Quantum computing and communication," *Adv. Comput.*, vol. 56, pp. 189–244, 2002.

[28] Eiben A.E. and Smith J.E., *Introduction to Evolutionary Computing*, Second. Springer, 2015.

[29] M. A. Hammer, K. Y. Phang, M. Hicks, and J. S. Foster, "Adapton: Composable, demand-driven incremental computation," 2014.

[30] I. J. Fister, X. S. Yand, I. Fister, J. Brest, and D. Fister, "A Brief Review of Nature-Inspired Algorithms for Optimization," *Electrotech. Rev.*, vol. 80, no. 3, 2013.

[31] K. F. Man, K. S. Tang, and S. Kwong, "Genetic algorithms: concepts and applications [in engineering design]," *IEEE Trans. Ind. Electron.*, vol. 43, no. 5, pp. 519–534, Oct. 1996, doi: 10.1109/41.538609.

[32] P. J. Bentley, T. G. W. Gordon, J. Kim, and S. Kumar, "New trends in evolutionary computation," 2001, vol. 1, pp. 162–169, doi: 10.1109/CEC.2001.934385.

[33] A. K. Kar, "Bio inspired computing – A review of algorithms and scope of applications," *Expert Syst. Appl.*, vol. 59, pp. 20–32, 2016, doi: https://doi.org/10.1016/j.eswa.2016.04.018.

[34] N. Siddique and H. Adeli, “Nature Inspired Computing: An Overview and Some Future Directions,” *Cogn. Comput.*, vol. 7, no. 6, pp. 706–714, Dec. 2015, doi: DOI 10.1007/s12559-015-9370-8.

[35] A. S. Karunananda, *Artificial intelligence*. Tharanji Prints, 2004.

[36] J. Liu and K. C. Tsui, “Toward nature-inspired computing,” *Commun. ACM*, vol. 49, no. 10, pp. 59–64, 2006.

[37] S. Costantini and G. De Gasperis, “Memory, experience and adaptation in logical agents,” in *Management Intelligent Systems*, Springer, 2013, pp. 17–24.

[38] B. Bakker, “Reinforcement learning with long short-term memory,” in *Advances in neural information processing systems*, 2002, pp. 1475–1482.

[39] F. C. Bartlett, F. C. Bartlett, and W. Kintsch, *Remembering: A study in experimental and social psychology*, vol. 14. Cambridge University Press, 1995.

[40] D. L. Schacter, “Constructive memory: past and future,” *Dialogues Clin. Neurosci.*, vol. 14, no. 1, p. 7, 2012.

[41] Kenneth J. Malmberg, Jeroen G. W. Raaijmakers, and Richard M. Shiffrin, “50 years of research sparked by Atkinson and Shiffrin (1968),” *Mem. Cognit.*, vol. 47, no. 4, pp. 561–574, 2019.

[42] A. Baddeley, “Working memory,” *Curr. Biol.*, vol. 20, no. 4, pp. R136–R140, 2010, doi: https://doi.org/10.1016/j.cub.2009.12.014.

[43] K. A. De Jong, *Evolutionary Computaion: A Unified Approach*. London, England: The MIT Press, Cambridge, Massachusetts, 2006.

[44] M. Sharma, P. Sindhwani, and V. Maheshwari, “Genetic Algorithm Optimal approach for Scheduling Processes in Operating System,” *Int. J. Comput. Sci. Netw. Secur.*, vol. 14, no. 5, pp. 91–94, 2014.

[45] Sindhwani P. and Wadhwa V., “Genetic algorithm Approach for Optimal CPU Scheduling,” *IJCST*, vol. 2, no. 2, pp. 92–95, Jun. 2011.

[46] M. U. Siregar, “A New Approach to CPU Scheduling: Genetic Round Robin,” *Int. J. Comput. Appl.*, vol. 47, no. 19, pp. 18–25, Jun. 2012.

[47] M. Carlsson, “Monads for incremental computing,” in *The seventh ACM SIGPLAN international conference on Functional programming*, 2002, pp. 26–35.

[48] Matthew A. Hammer and Umut A. Acar, “Memory Management for Self-Adjusting Computation,” presented at the Proceedings of the the 2008 International Symposium on Memory Management, Tucson, Arizona, USA, 2008.

[49] U. A. Acar, G. E. Blelloch, and R. Harper, *Selective memoization*, vol. 38. ACM, 2003.

[50] U. A. Acar, G. E. Blelloch, M. Blume, R. Happer, and K. Tangwongsan, “An experimental Analysis of Self-Adjusting Computation,” *ACM Trans. Program. Lang. Syst.*, vol. 32, no. 1, 2009.

[51] Xiaoming Li, Maria Jesus Grzaran, and David Padua, “A dynamically Tuned Sorting Lirary,” Los Alamitos, Calif., 2004.

[52] Y. A. Liu, “Efficient computation via incremental computation,” in *Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, 1998, vol. 1574, pp. 194–203.

[53] W. F. Jayasuriya, *The Psychology & Philosophy of Buddhism*. Onalaska, USA: pariyatti, 2016.

[54] R. C. N. Thero, *Pattana Maha Pakarana Sannaya*, 6th ed. Pokunuwita, Sri Lanka: Sri Chandravimala Darma P, 2014.

[55] B. Bodhi, *Comprehensive Manual of Abhidhamma: The Psychology of Buddhism (Abhidhammattha Sangaha)*. Buddhist Publication Society, 2006.

[56] M. Sipser, *Introduction to the Theory of Computation*, Second. Thomson Course Technology, 2007.

[57] Wong, H. S. P. and Salahuddin, S., "Memory leads the way to better computing," *Nat. Nanotechnol.*, vol. 10, no. 3, pp. 191–194, Mar. 2015.

[58] S. Jain, A. Ranjan, K. Roy, and A. Raghunathan, "Computing in memory with spin-transfer torque magnetic ram," *IEEE Trans. Very Large Scale Integr. VLSI Syst.*, vol. 26, no. 3, pp. 470–483, 2017.

[59] A. J. Smith, "Cache Memories," *ACM Comput. Surv.*, vol. 14, no. 3, pp. 473–530, Sep. 1982.

[60] W. J. Starke *et al.*, "The cache and memory subsystems of the IBM POWER8 processor," *IBM J. Res. Dev.*, vol. 59, no. 1, p. 3:1-3:13, Jan. 2015, doi: 10.1147/JRD.2014.2376131.

[61] LARRY HARDESTY, "Computer Chip Cores Communicate by Networks Instead of Bus," *SciTechDaily*, 2012. https://scitechdaily.com/computer-chip-cores-communicate-by-networks-instead-of-bus/.

[62] Kevin Lee, "Best RAM 2019: the top memory for your PC," *Techrader: Source for Tech buying Advices*, Oct. 04, 2019. https://www.techradar.com/news/best-ram.

[63] Scharon Harding, "What Is CAS Latency in RAM? CL Timings Explained," *Tom's Hardware*, Sep. 03, 2019. https://www.tomshardware.com/reviews/cas-latency-ram-cl-timings-glossary-definition,6011.html (accessed Oct. 26, 2019).

[64] S. Ghose, K. Hsieh, A. Boroumand, R. Ausavarungnirun, and O. Mutlu, "Enabling the adoption of processing-in-memory: Challenges, mechanisms, future research directions," *ArXiv Prepr. ArXiv180200320*, 2018.

[65] H. Asghari-Moghaddam, Y. H. Son, J. H. Ahn, and N. S. Kim, "Chameleon: Versatile and practical near-DRAM acceleration architecture for large memory systems," in *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, Taipei, Taiwan, Oct. 2016, pp. 1–13, doi: 10.1109/MICRO.2016.7783753.

[66] Mihir Patkar, "Intel Core i9 vs. i7 vs. i5: Which CPU Should You Buy?," *Technology Explained*, Dec. 06, 2018. https://www.makeuseof.com/tag/intel-core-i9-vs-i7-vs-i5-cpu/.

[67] J. D. Owens, M. Houston, D. Luebke, S. Green, J. E. Stone, and J. C. Phillips, "GPU computing," 2008.

[68] J. Nickolls and W. J. Dally, "The GPU computing era," *IEEE Micro*, vol. 30, no. 2, pp. 56–69, 2010.

[69] Y. Kim, D. Han, O. Mutlu, and M. Harchol-Balter, "ATLAS: A scalable and high-performance scheduling algorithm for multiple memory controllers," in *HPCA-16*

2010 The Sixteenth International Symposium on High-Performance Computer Architecture, 2010, pp. 1–12.

[70] Z. Guz, E. Bolotin, I. Keidar, A. Kolodny, A. Mendelson, and U. C. Weiser, “Many-core vs. many-thread machines: Stay away from the valley,” *IEEE Comput. Archit. Lett.*, vol. 8, no. 1, pp. 25–28, 2009.

[71] F. Gebali, *Algorithms and parallel computing*. Hoboken, N.J: Wiley, 2011.

[72] A. H. Almutairi and A. H. Alruwaili, “Improving of Quicksort Algorithm Performance by Sequential Thread or Parallel Algorithms,” *Glob. J. Comput. Sci. Technol. - Hardw. Comput.*, vol. 12, no. 10, 2012.

[73] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating system concepts*, Ninth edition. India: Wiley India Pvt. Ltd., 2016.

[74] Yanhong A. Liu, Scott D. Stoller, and Tim Teitelbaum, “Discovering auxiliary information for incremental computation,” in *POPL ’96 Proceedings of the 23rd ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, Florida, USA, Jan. 1996, pp. 157–170.

[75] Liu Y A, “CACHET: an interactive, incremental-attribution-based program transformation system for deriving incremental programs,” presented at the Proceedings 1995 10th Knowledge-Based Software Engineering Conference, 1995.

[76] M. A. Hammer *et al.*, “Incremental computation with names,” *ACM SIGPLAN Not.*, vol. 50, no. 10, pp. 748–766, Oct. 2015.

[77] Yanhong A. Liu, Scott D. Stoller, and Tim Teitelbaum, “Static caching for incremental computation,” *ACM Trans. Program. Lang. Syst. TOPLAS*, vol. 20, no. 3, pp. 546–585, May 1998.

[78] U. A. Acar, G. Blelloch, and R. Harper, “Self-adjusting computation,” PhD Thesis, Citeseer, 2005.

[79] U. A. Acar, A. Ahmed, and M. Blume, “Imperative Self-adjusting Computation,” *SIGPLAN Not*, vol. 43, no. 1, pp. 309–322, Jan. 2008, doi: 10.1145/1328897.1328476.

[80] P. Bhatotia, A. Wieder, R. Rodrigues, U. A. Acar, and R. Pasquin, “Incoop: MapReduce for incremental computations,” in *Proceedings of the 2nd ACM Symposium on Cloud Computing*, 2011, p. 7.

[81] M. A. Hammer, U. A. Acar, and Y. Chen, “CEAL: a C-based language for self-adjusting computation,” in *ACM Sigplan Notices*, 2009, vol. 44, pp. 25–37.

[82] P. Bhatotia, P. Fonseca, U. A. Acar, B. B. Brandenburg, and R. Rodrigues, “iThreads: A threading library for parallel incremental computation,” in *ACM SIGPLAN Notices*, 2015, vol. 50, pp. 645–659.

[83] K. Gharachorloo, D. Lenoski, J. Laudon, P. Gibbons, A. Gupta, and J. Hennessy, *Memory consistency and event ordering in scalable shared-memory multiprocessors*, vol. 18. ACM, 1990.

[84] A. S. Jensen and J. Villadsen, “A comparison of organization-centered and agent-centered multi-agent systems,” *Artif. Intell. Res.*, vol. 2, no. 3, Apr. 2013, doi: 10.5430/air.v2n3p59.

[85] N. R. Jannings, “An agent-based approach for building complex software systems,” *Coomunications ACM*, vol. 44, no. 4, pp. 35–41, Jan. 2001.

[86] S. Costantini, “Defining and Maintaining Agent’s Experience in Logical Agents,” in *Informal Proc. of the LPMAS (Logic Programming for Multi-Agent Systems) Workshop at ICLP 2011, and CORR Proceedings of LANMR 2011, Latin-American Conference on NonMonotonic Reasoning*, Mexico, 2011, pp. 151–165.

[87] Gero J. S., “Understanding behaviors of a constructive memory agent: A markov chain analysis.,” *Knowl.-Based Syst.*, vol. 22, no. 8, pp. 610–621, 2009.

[88] P.-S. Liew and J. S. Gero, “Constructive memory for situated design agents,” *AI EDAM*, vol. 18, no. 2, pp. 163–198, 2004.

[89] Xiaoming Li, Maria Jesus Grzaran, and David Padua, “Optimizing sorting with genetic algorithms,” New York, NY, USA., 2005, doi: 10.1109/CGO.2005.24.

[90] L. Daniel and K. T. Chaturvedi, “Review on Different Evolutionary Computing Techniques In Particle swarm Optimization,” *Int. J. Eng. Sci. Math.*, vol. 7, no. 3, pp. 230–236, 2018.

[91] A. Ahilan and P. Deepa, “Improving lifetime of memory devices using evolutionary computing based error correction coding,” in *Computational intelligence, cyber security and computational models*, Springer, 2016, pp. 237–245.

[92] Gunther Raidl, “Evolutionary computation: An overview and recent trends,” *OGAI*, pp. 2–7, 2005.

[93] P. de Wilde and G. Briscoe, "“Stability of evolving multiagent systems,” *” IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 41, no. 4, pp. 1149–1157, 2011.

[94] C. Tacla and J.-P. Barthès, “A multi-agent architecture for evolving memories,” 2003.

[95] M. C. Mozer and others, “Neural net architectures for temporal sequence processing,” in *Santa Fe Institute Studies in the Sciences of Complexity-Proceedings Volume-*, 1993, vol. 15, pp. 243–243.

[96] M. D. Godfrey and D. F. Hendry, “The computer as von Neumann planned it,” *IEEE Ann. Hist. Comput.*, vol. 15, no. 1, pp. 11–21, 1993, doi: 10.1109/85.194088.

[97] L. Wang *et al.*, “Superneurons: dynamic GPU memory management for training deep neural networks,” in *ACM SIGPLAN Notices*, 2018, vol. 53, pp. 41–53.

[98] P. Chi *et al.*, “Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory,” in *ACM SIGARCH Computer Architecture News*, 2016, vol. 44, pp. 27–39.

[99] S. Khadka, J. J. Chung, and K. Tumer, “Evolving memory-augmented neural architecture for deep memory problems,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2017, pp. 441–448.

[100] A. Graves, G. Wayne, and I. Danihelka, “Neural turing machines,” *ArXiv Prepr. ArXiv14105401*, 2014.

[101] J. J. A. Ansel, “Autotuning programs with algorithmic choice,” PhD Thesis, Massachusetts Institute of Technology, 2014.

[102] P. Pfaffe, M. Tillmann, S. Walter, and W. F. Tichy, “Online-autotuning in the presence of algorithmic choice,” in *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2017, pp. 1379–1388.

[103] Y. Ding, J. Ansel, K. Veeramachaneni, X. Shen, U.-M. O’Reilly, and S. Amarasinghe, “Autotuning algorithmic choice for input sensitivity,” in *ACM SIGPLAN Notices*, 2015, vol. 50, pp. 379–390.

[104] J. Ansel *et al.*, “Opentuner: An extensible framework for program autotuning,” in *Proceedings of the 23rd international conference on Parallel architectures and compilation*, 2014, pp. 303–316.

[105] J. Ansel, Y. L. Wong, C. Chan, M. Olszewski, A. Edelman, and S. Amarasinghe, “Language and compiler support for auto-tuning variable-accuracy algorithms,” in *International Symposium on Code Generation and Optimization (CGO 2011)*, 2011, pp. 85–96.

[106] J. Ansel *et al.*, “Siblingrivalry: online autotuning through local competitions,” in *Proceedings of the 2012 international conference on Compilers, architectures and synthesis for embedded systems*, 2012, pp. 91–100.

[107] W.-C. Lee *et al.*, “White-box program tuning,” in *Proceedings of the 2019 IEEE/ACM International Symposium on Code Generation and Optimization*, 2019, pp. 122–135.

[108] P. Balaprakash *et al.*, “Autotuning in High-Performance Computing Applications,” *Proc. IEEE*, vol. 106, no. 11, pp. 2068–2083, 2018.

[109] P. Tillet and D. Cox, “Input-aware auto-tuning of compute-bound HPC kernels,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2017, p. 43.

[110] Chen, G, Church, D. A, Englert, B. G., and Zubairy, M. S., “Mathematical models of contemporary elementary quantum computing devices,” in *Quantum Control: Mathematical and Numerical Challenges*, 2003, vol. 33, pp. 79–117.

[111] Calude C. S and Michel J. D., “Reflections on Quantum Computing,” 2000.

[112] Federman A., “What Buddhism Taught Cognitive Science about Self, Mind and Brain,” *Enrahonar Quad. Filos.*, vol. 47, pp. 39–62, 2011.

[113] Nàrada Mahà Thera, *A Manual of Abhidhamma (Abhidhammattha Saïgaha of Bhadanta Anuruddhàcariya)*. Malaysia: Biddhist Missionary Society, 1987.

[114] Nayanaponika Thero, *Abhidhamma Studies: Researches in Buddhist Psycology*. Kandy, Sri Lanka: Buddhist Publication Society, 1976.

[115] Bhaddanta Dr. Rewata Dhamma, *Process of Consiousness and Matter*. Onalaska, USA: Pariyatti, 2015.

[116] A. Rathnapala, *Abhidharmartha Pradeepika*, vol. 3. Sri Lanka: M. D. Gunasena, 1973.

[117] Rerukane Chandrawimala Nayaka Thero, *Abhidharma Mulika Karunu*, 12th ed. 2009.

[118] Nārada and Pali Text Society, *Conditional relations (Paṭṭhāna)*. Bristol: Pali Text Society, 2010.

[119] “Suwisi Pratya - Pattana Prakarana,” in *Abhidhamma Pritaka, Buddha Jayanthi Tipitaka Series*, vol. 52–3, 2006.

[120] Pinker Steven, “How the mind works,” *Ann. N. Y. Acad. Sci.*, vol. 882, no. 1, pp. 119–127, 1999.

[121] Litch Mary, “Computation, connectionism and modelling the mind,” *Philos. Psychol.*, vol. 10, no. 3, p. 357, 1997.

[122] A. V. Aho, “Computation and Computational Thinking,” *Comput. J.*, vol. 55, no. 7, pp. 832–835, Jul. 2012, doi: 10.1093/comjnl/bxs074.

[123] Vallverdu J, “The Eastern Construction of the Artificial Mind,” *Enrahonar Quardens Filasofia 47*, pp. 171–185, 2011.

[124] Elodzislaw Duch, “What constitutes a good theory of mind,” *Am. Neopragmatism Torun*, vol. 8, pp. 8–12, 1998.

[125] Zhang J., “Memory Process and the Function of sleep,” *J. Theor.*, vol. 6, no. 6, 2004.

[126] D. E. Nee and M. D’Esposito, “Working Memory,” in *Brain Mapping*, A. W. Toga, Ed. Waltham: Academic Press, 2015, pp. 589–595.

[127] P. O. A. Haikonen, “The Challenges for Implementable Theories of Mind,” *J. Mind Theory*, vol. 0, no. 1, pp. 99–110, 2008.

[128] W. A. C. Weerakoon, A. S. Karunananda, and N. G. J. Dias, “Enhancing the Functionality of Rule-Based Expert Systems,” Kuala Lumpur, Malaysia, Sep. 2015, p. 93.

[129] T. J. Cleophas and A. H. Zwinderman, *SPSS for starters and 2nd levelers*, 2. ed. Heidelberg: Springer, 2015.

[130] G. E. Meek, C. Ozgur, and K. Dunning, “Comparison of the t vs. Wilcoxon Signed-Rank Test for Likert Scale Data and Small Samples,” vol. 6, no. 1, pp. 91–106, May 2007.

[131] A. Imam, U. Mohammed, and C. Moses Abanyam, “On Consistency and Limitation of paired t-test, Sign and Wilcoxon Sign Rank Test,” vol. 10, no. 1 Ver.IV, pp. 01–06, Feb. 2014.

[132] S. C. Gupta and V. K. Kapoor, *Fundamentals of Mathematical Statistics*. Sultan Chand & Sons, 2007.

[133] R. E. Walpole, *Probability and statistics for engineers and scientists*. Boston, Mass: Pearson Education, 2012.

[134] Jim Colton, Minitab Inc., *Detecting and Analyzing Non-Normal data*. Florida, 2014.

[135] M. A. Stephen, “The Anderson-Darlin Statistics,” DEPARTMENT OF STATISTICS STANFORD UNIVERSITY STANFORD, CALIFORNIA, 1979.

[136] F. E. Grubbs, “Procedures for Detecting Outlying Observations in Samples,” *Am. Stat. Assoc. Am. Soc. Qual.*, vol. 11, no. 1, pp. 1–21, Feb. 1969.

[137] H. J. Hopcroft, R. Motwani, and D. J. Ullman, *Introduction to Automata Theory Languages, and Computation*, Third. Pearson Education, 2007.

[138] A. T. Sudkamp, *Languages and Machines: An Introduction to the Theory of Computer Science*, Third. Pearson, 2005.

[139] C. J. Martin, *Introduction to Languages and the Theory of Computation*, 2007th ed. McGraw Hill Education, 2014.

[140] W. A. C. Weerakoon, A. S. Karunananda, and N. G. J. Dias, “New Processing Model for Operating Systems,” University of Kelaniya, 2016, p. 29, [Online]. Available: http://repository.kln.ac.lk/handle/123456789/15931.