Table 4
Impact of Gaussian noise on the accuracy.

Iteration	Percentage Noise $\alpha_n = 5$			Percentage Noise $\alpha_n = 2$		
	η_{nl}	ϵ_l	$\tau_i(x)$	η_{nl}	ϵ_l	$\tau_i(x)$
PSO	6.1	0.69	0.73	6.3	0.65	0.56
ICSO	5.2	0.44	0.68	5.5	0.38	0.47

Table 5
Mean localization error.

Algorithm	Mean
RSSI	0.551
PSO	0.321
ICSO	0.319

The beacon nodes are located at the four corners of the rectangle region, as shown in Fig. 14. Fig. 15 depicts the averaged RSS distances between the beacon and the target node in indoor environments.

Error analysis

RSSI-based distance measurement exhibits more variation due to the effects of diminishing and shadowing. Because of the presence of walls and other metallic objects, there were more reflections indoors. Because each reflected signal takes a different path with a different amplitude and phase. The PSO and ICSO algorithms can significantly reduce the mean position error. In an indoor environment, the RSSI-based method provides distance estimation with an average error of 0.713 m. The ICSO algorithm is used to reduce the localization error to 0.423 m, whereas the PSO algorithm offers 0.597 m. The position error computed by the RSSI, PSO, and ICSO algorithms is depicted in Fig. 16. Because of its superior global searching efficiency, the heuristic algorithm ICSO has been found to help reduce localization errors over PSO. Table 5 displays the mean and standard deviation values of position error computed by the proposed and existing algorithms.

5. Conclusion & future scope

Localization is an important performance issue in the sustainable IIoT, as it is critical in energy-efficient cluster protocols. The main goal is to use bio-inspired algorithms to reduce localization errors. A measurement system with a low-power transceiver and microcontroller is designed in this paper to realize both PSO and ICSO algorithms in the optimization of node localization for IIoT. The RSS indicator method is used to calculate the relative distance between the target node and its receiver. However, it has been discovered that the RSS indicator technique is not very accurate and is influenced by the indoor environment. In terms of position accuracy, the experimental results show that PSO and ICSO outperform the simple RSSI technique. However, the ICSO has a slight advantage over the PSO and, due to its global searching strategy, requires less computation time. The simulated results also show that the ICSO algorithm outperforms the PSO algorithm in terms of localization error and computing time. The effects of additive Gaussian noise, beacon density, target node densities, and communication radius on localization accuracy were also investigated. Furthermore, the goal is to use the algorithms for centralized energy-aware clustering to reduce energy consumption in sustainable IIoT networks and green cities. Furthermore, the ICSO can be combined with other evolutionary algorithms to reduce location estimation errors.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data

References

Abdulzahra, A. M. K., Al-Qurabat, A. K. M., & Abdulzahra, S. A. (2023). Optimizing energy consumption in WSN-based IoT using unequal clustering and sleep scheduling methods. *Internet of Things*, Article 100765.

Arya, R. (2022). Exploiting perturbed and coalescent anchor node geometry with semidefinite relaxation for sensor network localization. *Physical Communication*, *52*, Article 101606.

Behera, T. M., Mohapatra, S. K., Samal, U. C., & Khan, M. S. (2019). Hybrid heterogeneous routing scheme for improved network performance in WSNs for animal tracking. *Internet of Things*, *6*, Article 100047.

Behera, T. M., Mohapatra, S. K., Samal, U. C., Khan, M. S., Daneshmand, M., & Gandomi, A. H. (2019). Residual energy-based cluster-head selection in WSNs for IoT application. *IEEE Internet of Things Journal*, *6*(3), 5132–5139.

Chen, S., Zhang, L., Tang, Y., Shen, C., Kumar, R., Yu, K., et al. (2020). Indoor temperature monitoring using wireless sensor networks: A SMAC application in smart cities. *Sustainable Cities and Society*, *61*, Article 102333.

Cheng, J., Li, Y., & Xu, Q. (2022). An anchor node selection scheme for improving RSS-based localization in wireless sensor network. *Mobile Information Systems*, *2022*.

Chithaluru, P. (2020). *Energy efficient routing approach based on volunteer participation and adaptive ranking of forwarder nodes in WSNs* (Doctoral dissertation), UPES, Dehradun: School of Computer Science.

Chithaluru, P., Al-Turjman, F., Kumar, M., & Stephan, T. (2020). I-AREOR: An energy-balanced clustering protocol for implementing green IoT in smart cities. *Sustainable Cities and Society*, *61*, Article 102254.

Chithaluru, P., Al-Turjman, F., Kumar, M., & Stephan, T. (2021). MTCEE-LLN: Multilayer threshold cluster-based energy-efficient low-power and lossy networks for industrial Internet of Things. *IEEE Internet of Things Journal*, *9*(7), 4940–4948.

Chithaluru, P., Al-Turjman, F., Kumar, M., & Stephan, T. (2023). Energy-balanced neuro-fuzzy dynamic clustering scheme for green & sustainable IoT based smart cities. *Sustainable Cities and Society*, Article 104366.

Chithaluru, P., Al-Turjman, F., Stephan, T., Kumar, M., & Mostarda, L. (2021). Energy-efficient blockchain implementation for Cognitive Wireless Communication Networks (CWCNs). *Energy Reports*, *7*, 8277–8286.

Chithaluru, P., Fadi, A. T., Kumar, M., & Stephan, T. (2023). Computational intelligence inspired adaptive opportunistic clustering approach for industrial IoT networks. *IEEE Internet of Things Journal*.

Chithaluru, P., Jena, L., Patra, B., & Panda, N. (2022). Enhanced machine learning algorithms for validating the biometrics in VANET. In *2022 international conference on machine learning, computer systems and security* (pp. 11–16). IEEE.

Chithaluru, P., Jena, L., Singh, D., & Nayak, S. (2022). Statistical validations on energy efficient routing protocols to improve the performance of wireless sensor network. In *2022 international conference on machine learning, computer systems and security* (pp. 325–330). IEEE.

Chithaluru, P., Jena, L., Singh, D., & Ravi Teja, K. M. V. (2022). An adaptive fuzzy-based clustering model for healthcare wireless sensor networks. In *Ambient intelligence in health care: proceedings of ICAIHC 2022* (pp. 1–10). Singapore: Singapore Springer Nature.

Chithaluru, P. K., Khan, M. S., Kumar, M., & Stephan, T. (2021). ETH-LEACH: An energy enhanced threshold routing protocol for WSNs. *International Journal of Communication Systems*, *34*(12), Article e4881.

Chithaluru, P., Kumar, S., Singh, A., Benslimane, A., & Jangir, S. K. (2021). An energy-efficient routing scheduling based on fuzzy ranking scheme for Internet of Things (IoT). *IEEE Internet of Things Journal*.

Chithaluru, P., & Prakash, R. (2018). Simulation on SDN and NFV models through mininet. In *Innovations in software-defined networking and network functions virtualization* (pp. 149–174). IGI Global.

Chithaluru, P., & Prakash, R. (2020). Organization security policies and their after effects. In *Information security and optimization* (pp. 43–60). Boca Raton London New York: CRC Press.

Chithaluru, P., Prakash, R., & Srivastava, S. (2018). WSN structure based on SDN. In *Innovations in software-defined networking and network functions virtualization* (pp. 240–253). IGI Global.

Chithaluru, P., Singh, A., Mahmoud, M. S., Kumar, S., Mazón, J. L. V., Alkhayyat, A., et al. (2023). An enhanced opportunistic rank-based parent node selection for sustainable & smart IoT networks. *Sustainable Energy Technologies and Assessments*, *56*, Article 103079.

Chithaluru, P., Singh, K., & Sharma, M. K. (2020). Cryptocurrency and blockchain. (pp. 143–158). Boca Raton London New York: CRC Press,

Chithaluru, P., Stephan, T., Kumar, M., & Nayyar, A. (2022). An enhanced energy-efficient fuzzy-based cognitive radio scheme for IoT. *Neural Computing and Applications*, *34*(21), 19193–19215.

Chithaluru, P., Tanwar, R., & Kumar, S. (2020). Cyber-attacks and their impact on real life: What are real-life cyber-attacks, how do they affect real life and what should we do about them? *Information Security and Optimization*, *61*.

Chithaluru, P., Tiwari, R., & Kumar, K. (2019). AREOR–Adaptive ranking based energy efficient opportunistic routing scheme in wireless sensor network. *Computer Networks*, *162*, Article 106863.

Chithaluru, P., Tiwari, R., & Kumar, K. (2021a). Arior: Adaptive ranking based improved opportunistic routing in wireless sensor networks. *Wireless Personal Communications*, *116*(1), 153–176.

Chithaluru, P., Tiwari, R., & Kumar, K. (2021b). Performance analysis of energy efficient opportunistic routing protocols in wireless sensor network. *International Journal of Sensors Wireless Communications and Control*, *11*(1), 24–41.

Ding, P., Holliday, J., & Celik, A. (2005). Distributed energy-efficient hierarchical clustering for wireless sensor networks. In *International conference on distributed computing in sensor systems* (pp. 322–339). Berlin, Heidelberg: Springer.

Guru, S. M., Halgamuge, S. K., & Fernando, S. (2005). Particle swarm optimisers for cluster formation in wireless sensor networks. In *2005 international conference on intelligent sensors, sensor networks and information processing* (pp. 319–324). IEEE.

Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communication*, *1*(4), 660–670.

Jain, A., Singh, J., Kumar, S., Ţ, Florin-Emilian, Traian Candin, M., & Chithaluru, P. (2022). Improved recurrent neural network schema for validating digital signatures in VANET. *Mathematics*, *10*(20), 3895.

Jena, L., Ammoun, L., & Chithaluru, P. (2022). Supervised intelligent clinical approach for breast cancer tumor categorization. In *Augmented intelligence in healthcare: A pragmatic and integrated analysis* (pp. 15–40). Singapore: Singapore, Springer Nature.

Jia, W., Qi, G., Liu, M., & Zhou, J. (2022). A high accuracy localization algorithm with DV-Hop and fruit fly optimization in anisotropic wireless networks. *Journal of King Saud University-Computer and Information Sciences*.

Joshi, D., Ali Albahar, M., Chithaluru, P., Singh, A., Yadav, A., & Miro, Y. (2022). A novel approach to integrating uncertainty into a push re-label network flow algorithm for pit optimization. *Mathematics*, *10*(24), 4803.

Joshi, D., Chithaluru, P., Anand, D., Hajjej, F., Aggarwal, K., Torres, V. Y., et al. (2023). An evolutionary technique for building neural network models for predicting metal prices. *Mathematics*, *11*(7), 1675.

Joshi, D., Chithaluru, P., Singh, A., Yadav, A., Elkamchouchi, D. H., Breñosa, J., et al. (2022). An optimized open pit mine application for limestone quarry production scheduling to maximize net present value. *Mathematics*, *10*(21), 4140.

Joshi, D., Chithaluru, P., Singh, A., Yadav, A., Elkamchouchi, D. H., Pérez-Oleaga, C. M., et al. (2022). A novel large-scale stochastic pushback design merged with a minimum cut algorithm for open pit mine production scheduling. *Systems*, *10*(5), 159.

Kulkarni, R. V., & Venayagamoorthy, G. K. (2010). Particle swarm optimization in wireless-sensor networks: A brief survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *41*(2), 262–267.

Lashkari, B., Rezazadeh, J., Farahbakhsh, R., & Sandrasegaran, K. (2018). Crowdsourcing and sensing for indoor localization in IoT: A review. *IEEE Sensors Journal*, *19*(7), 2408–2434.

Liu, W., Luo, X., Wei, G., & Liu, H. (2022). Node localization algorithm for wireless sensor networks based on static anchor node location selection strategy. *Computer Communications*, *192*, 289–298.

Loscri, V., Morabito, G., & Marano, S. (2005). A two-levels hierarchy for low-energy adaptive clustering hierarchy (TL-LEACH). In *IEEE vehicular technology conference, vol. 62, no. 3* (p. 1809). IEEE, 1999.

Luomala, J., & Hakala, I. (2022). Adaptive range-based localization algorithm based on trilateration and reference node selection for outdoor wireless sensor networks. *Computer Networks*, *210*, Article 108865.

Mohar, S. S., Goyal, S., & Kaur, R. (2022). Localization of sensor nodes in wireless sensor networks using bat optimization algorithm with enhanced exploration and exploitation characteristics. *The Journal of Supercomputing*, *78*(9), 11975–12023.

Niculescu, V., Palossi, D., Magno, M., & Benini, L. (2022). Fly, wake-up, find: UAV-based energy-efficient localization for distributed sensor nodes. *Sustainable Computing: Informatics and Systems*, *34*, Article 100666.

Ou, X., Wu, M., Pu, Y., Tu, B., Zhang, G., & Xu, Z. (2022). Cuckoo search algorithm with fuzzy logic and Gauss–Cauchy for minimizing localization error of WSN. *Applied Soft Computing*, *125*, Article 109211.

Pradhan, A. K., Das, K., Mishra, D., & Chithaluru, P. (2023). Optimizing CNN-LSTM hybrid classifier using HCA for biomedical image classification. *Expert Systems*, Article e13235.

Prakash, R., & Chithaluru, P. (2021). Active security by implementing intrusion detection and facial recognition. In *Nanoelectronics, circuits and communication systems* (pp. 1–7). Singapore: Springer.

Prakash, R., Chithaluru, P., Sharma, D., & Srikanth, P. (2019). Implementation of trapdoor functionality to two-layer encryption and decryption by using RSA-AES cryptography algorithms. In *Nanoelectronics, circuits and communication systems* (pp. 89–95). Springer: Singapore.

Qin, T., Li, P., & Shen, S. (2018). Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, *34*(4), 1004–1020.

Ramakuri, S. K., Chithaluru, P., & Kumar, S. (2019). Eyeblink robot control using brain-computer interface for healthcare applications. *International Journal of Mobile Devices, Wearable Technology, and Flexible Electronics (IJMDWTFE)*, *10*(2), 38–50.

Sabale, K., & Mini, S. (2022). Path planning mechanism for mobile anchor-assisted localization in wireless sensor networks. *Journal of Parallel and Distributed Computing*, *165*, 52–65.

Sasikumar, P., & Khara, S. (2012). K-Means clustering in wireless sensor networks. In *2012 fourth international conference on computational intelligence and communication networks* (pp. 140–144). IEEE.

Shah, S. B., Zhe, C., Yin, F., Khan, I. U., Begum, S., Faheem, M., et al. (2018). 3D weighted centroid algorithm & RSSI ranging model strategy for node localization in WSN based on smart devices. *Sustainable Cities and Society*, *39*, 298–308.

Shit, R. C., Sharma, S., Puthal, D., & Zomaya, A. Y. (2018). Location of Things (LoT): A review and taxonomy of sensors localization in IoT infrastructure. *IEEE Communications Surveys & Tutorials*, *20*(3), 2028–2061.

Sridhar, P. S. V. S., Chithaluru, P., Kumar, S., Cheikhrouhou, O., & Hamam, H. (2023). An enhanced haar cascade face detection schema for gender recognition. In *2023 international conference on smart computing and application* (pp. 1–5). IEEE.

Tanwar, R., Balamurugan, S., Saini, R. K., Bharti, V., & Chithaluru, P. (Eds.), (2022). *Advanced healthcare systems: empowering physicians with IoT-enabled technologies*. John Wiley & Sons.

Tillett, J., Rao, R., & Sahin, F. (2002). Cluster-head identification in ad hoc sensor networks using particle swarm optimization. In *2002 IEEE international conference on personal wireless communications* (pp. 201–205). IEEE.

Wu, Z. Y., Chew, A., Meng, X., Cai, J., Pok, J., Kalfarisi, R., et al. (2023). High fidelity digital twin-based anomaly detection and localization for smart water grid operation management. *Sustainable Cities and Society*, *91*, Article 104446.

Xu, J., Jin, N., Lou, X., Peng, T., Zhou, Q., & Chen, Y. (2012). Improvement of LEACH protocol for WSN. In *2012 9th international conference on fuzzy systems and knowledge discovery* (pp. 2174–2177). IEEE.

Yadav, A., Albahar, M. Ali., Chithaluru, P., Singh, A., Alammari, A., Kumar, G. V., et al. (2023). Hybridizing artificial intelligence algorithms for forecasting of sediment load with multi-objective optimization. *Water*, *15*(3), 522.

Yadav, A., Chithaluru, P., Singh, A., Albahar, M. A., Jurcut, A., Álvarez, R. M., et al. (2022). Suspended sediment yield forecasting with single and multi-objective optimization using hybrid artificial intelligence models. *Mathematics*, *10*(22), 4263.

Yadav, A., Chithaluru, P., Singh, A., Joshi, D., Elkamchouchi, D. H., Pérez-Oleaga, C. M., et al. (2022). An enhanced feed-forward back propagation Levenberg–Marquardt algorithm for suspended sediment yield modeling. *Water*, *14*(22), 3714.

Yan, Y., Yang, G., Wang, H., & Shen, X. (2022). Robust multiple sensor localization via semidefinite relaxation in wireless sensor networks with anchor position uncertainty. *Measurement*, *196*, Article 111193.

Younis, O., & Fahmy, S. (2004). HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, *3*(4), 366–379.

Zhang, D. G., Zhang, T., Zhang, J., Dong, Y., & Zhang, X. D. (2018). A kind of effective data aggregating method based on compressive sensing for wireless sensor network. *EURASIP Journal on Wireless Communications and Networking*, *2018*(1), 1–15.

Research Article

Mukkamula Venu Gopalachari*, Sangeeta Gupta, Salakapuri Rakesh, Dharmana Jayaram, and Pulipati Venkateswara Rao

Aspect-based sentiment analysis on multi-domain reviews through word embedding

https://doi.org/10.1515/jisys-2023-0001
received January 03, 2023; accepted April 27, 2023

Abstract: The finest resource for consumers to evaluate products is online product reviews, and finding such reviews that are accurate and helpful can be difficult. These reviews may sometimes be corrupted, biased, contradictory, or lacking in detail. This opens the door for customer-focused review analysis methods. A method called "Multi-Domain Keyword Extraction using Word Vectors" aims to streamline the customer experience by giving them reviews from several websites together with in-depth assessments of the evaluations. Using the specific model number of the product, inputs are continuously grabbed from different e-commerce websites. Aspects and key phrases in the reviews are properly identified using machine learning, and the average sentiment for each keyword is calculated using context-based sentiment analysis. To precisely discover the keywords in massive texts, word embedding data will be analyzed by machine learning techniques. A unique methodology developed to locate trustworthy reviews considers several criteria that determine what makes a review credible. The experiments on real-time data sets showed better results compared to the existing traditional models.

keywords: aspect-based sentiment analysis, product reviews, cold start, sentiment analysis, word embedding

1 Introduction

It could require a lot of time to conduct online research for product reviews [1]. A consumer must read multiple reviews on various websites to get a feel of a product's benefits and drawbacks. Customers are looking for a more accurate breakdown of all reviews when they read many testimonials on social media. Making the user's decision-making process easier would be to provide dependable reviews based on specific criteria. Finding a method for effectively condensing review data would not only assist customers in making wiser judgments but also increase market awareness of quality [2].

In addition, an efficient review analyzer would provide quick input that could be applied to improving services. As a result, the market needs quality keyword extraction and polarity quantifying techniques that would help in the optimal mapping of customers and companies. The use of sentiment analysis is advantageous in a wide range of different industries to sort out business challenges [3]. The main applications

*** Corresponding author: Mukkamula Venu Gopalachari,** Chaitanya Bharathi Institute of Technology, Hyderabad 500075, Telangana, India, e-mail: venugopal.m07@gmail.com
Sangeeta Gupta: Chaitanya Bharathi Institute of Technology, Hyderabad 500075, Telangana, India, e-mail: sangeetagupta_cse@cbit.ac.in
Salakapuri Rakesh: Chaitanya Bharathi Institute of Technology, Hyderabad 500075, Telangana, India, e-mail: srakesh_it@cbit.ac.in
Dharmana Jayaram: Chaitanya Bharathi Institute of Technology, Hyderabad 500075, Telangana, India, e-mail: djayaram_it@cbit.ac.in
Pulipati Venkateswara Rao: VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad 500075, Telangana, India, e-mail: pvenket2004@gmail.com

include market analysis, consumer opinion analysis, and the analysis of product reviews. They can be applied to many other sectors, such as e-commerce, services, and electronics. E-commerce platforms like Amazon and Flipkart have tried to reflect a product's features, but the existing techniques are frequently insufficient and lack the weight of more significant criteria because they only employ evaluations from the relevant industry [4]. The fewer data accessible in terms of size or users causes an issue known as "cold start," which has an impact on the quality of information retrieval [5]. The dynamics of the sentiment analysis over aspects in short text messages or reviews are more sensitive toward a single data source and thus get affected by the volume of the relevant part of the data. For instance, if the volume of reviews related to a particular item in e-commerce is low, it leads to a cold start, which in turn affects the quality of the aspects in sentiment analysis. The field of aspect-based sentiment analysis has had numerous different modifications and crossed many different new eras, thus it is not a straight journey. Researchers have been putting forth a lot of effort to address complex problems with numerous facets in the field of sentiment analysis. With a variety of machine-learning techniques, primarily deep-learning techniques, they have developed comprehensive answers to numerous complex issues with respect to the quality of aspects [6].

In this work, the proposed system develops a framework that effectively gathers product reviews from various e-commerce websites (e.g., Amazon, Flipkart, and Snapdeal) that determine the attributes of the concerned product along with their related adjectives and check for any other structural text discrepancies. In addition, a system to effectively gather product reviews from all three major e-commerce websites is developed, which extracts the features of the products discussed along with their corresponding adjectives while ensuring that there are no other structural text inconsistencies [5].

Aspect extraction (AE), aspect sentiment analysis, and sentiment evolution are the three basic processing phases that can be used to categorize aspect-based sentiment analysis (ABSA) as shown in Figure 1. The extraction of aspects, including explicit aspects, implicit aspects, aspect terms, entities, and opinion target expressions, is the focus of the first phase. The second stage categorizes the polarity of sentiment for a chosen aspect, target, or object. To increase the accuracy of sentiment classification, this phase also formulates interactions, dependencies, and contextual semantic linkages between various data items, such as aspect, entity, target, multi-word target, and sentiment word. Ternary, or fine-gained sentiment, values can be used to categorize the conveyed emotion. The third stage focuses on how the attitudes of people toward certain characteristics (or events) change over time, and sentiment evolution is thought to be mostly caused by social factors and personal experiences [7].

2 Related work

According to studies, SA may often be divided into three levels. To categorize whether a document as a whole, a statement (subjective or objective), or an aspect reflect a feeling, i.e., whether it is positive, negative, or neutral. Comparatively, the ABSA, which places a direct emphasis on sentiments rather than language structure, aids in a better understanding of the SA issue. The core idea of an aspect extends beyond judgment to include thoughts, points of view, ways of thinking, viewpoints, an underlying theme, or a social effect on an occurrence when an aspect is associated with an entity. Hence, ABSA offers a fantastic opportunity to analyze public opinions over time across various media-presented topics. Prior approaches have used a wide range of techniques to assist with the AE component, including parsing, named entity recognizers, bag-of-words, semantic analysis, and domain-specific ones like word clusters [8]. In addition, there are some techniques that focus on identifying the nouns that a viewpoint describes. However, word vectors, which convert words into vectors of a preset length, can be used to efficiently handle issues like the multi-class categorization of words [9]. When word vectors are used, the game changes as there are so many applications. In this work, it is discovered that the most accurate results were produced when this method was used with *K*-Means clustering.

Many strategies have been put forth in the literature for ABSA, including deep learning-based mode and more conventional feature-based models [10,11]. A few researchers attempted to apply the pre-trained bidirectional encoder representations from transformers model to ABSA in the context of the recent pattern of fine-tuning pre-

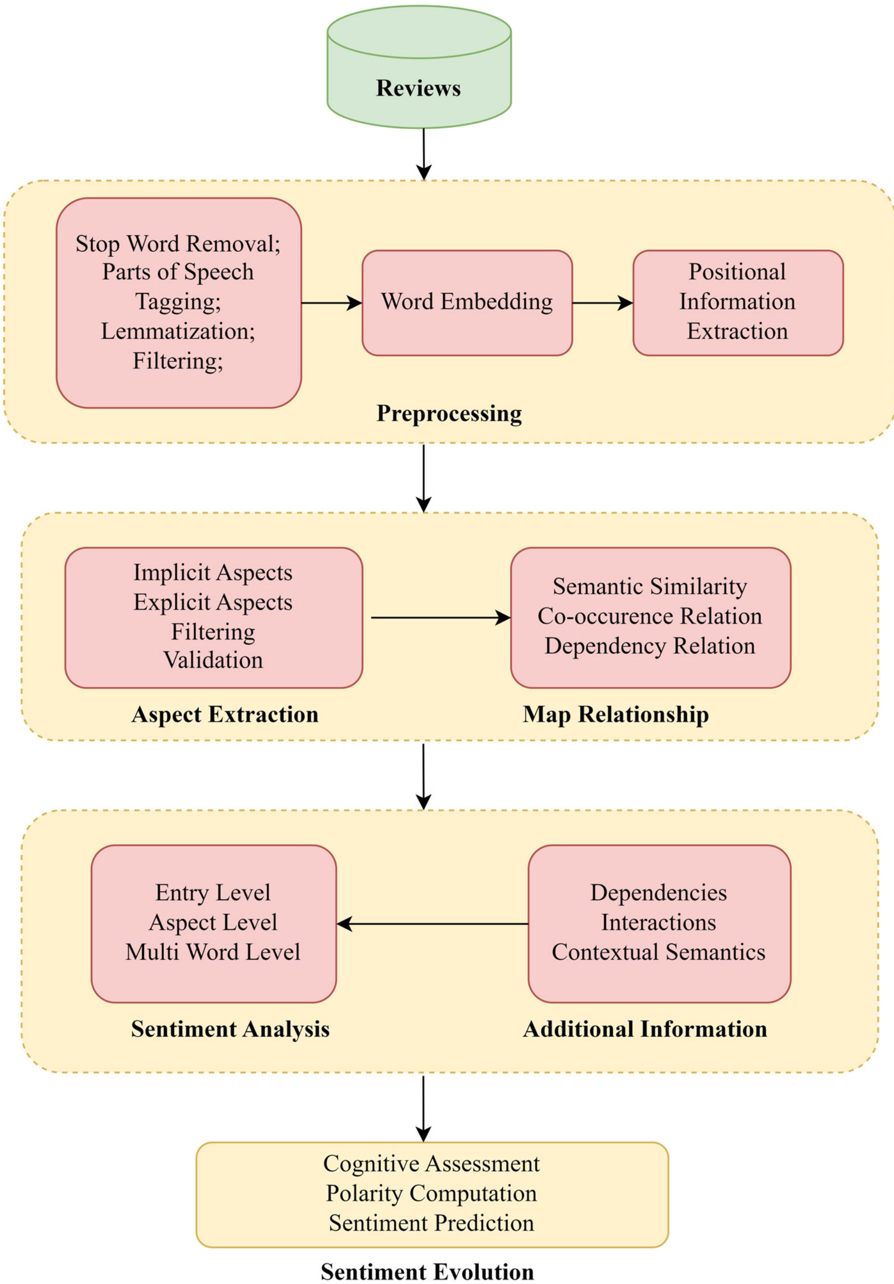

Figure 1: Traditional phases in aspect-based sentiment analysis.

trained models in natural language processing tasks, and they achieved the most cutting-edge results on multiple benchmark datasets [12]. Despite the substantial improvement, most of these systems mainly rely on textual content and ignore other related modalities, such as images with facial expressions. The input from various methods is crucial for anticipating the sentiment polarities regarding target aspects, as many online forums are becoming more multimodal. This encouraged a number of recent research to propose using beneficial information from images to increase the task performance of ABSA [13].

The phases of the review analysis process that can be separated are AE and multi-domain scraping [14]. The older techniques for extracting online reviews retrieved them from a single website. Multi-domain scraping has trouble identifying products and finding reviews that are hard to find using simple HTML parsers [15,16]. Either single domain scraping or multi domain scraping can handle browsing tasks are usually automated with tools like selenium will use the unique identifiers to search for products will resolve the mentioned concerns.

The accuracy may be further enhanced by recurrent neural network, which handles the context of phrase structures that our suggested system might not be able to recognize [17]. Reviews that are deemed insufficiently trustworthy are filtered using the trustability scores that have been calculated as a threshold. To gain the most reliable perspective on the product, the customer can also consider the reviews with the highest trustability rankings. The user receives a significantly more reliable appraisal of the product and its features because the least reliable scores are taken away [18].

3 Methodology

In order to avoid the cold start problem, the proposed system is modeled in two phases. In the first phase, dynamic multi-domain scraping is done, and in the second phase, product review keyword extraction and trustability scores are determined. The model in Figure 2 depicts the entire system design, which includes the aforementioned phases.

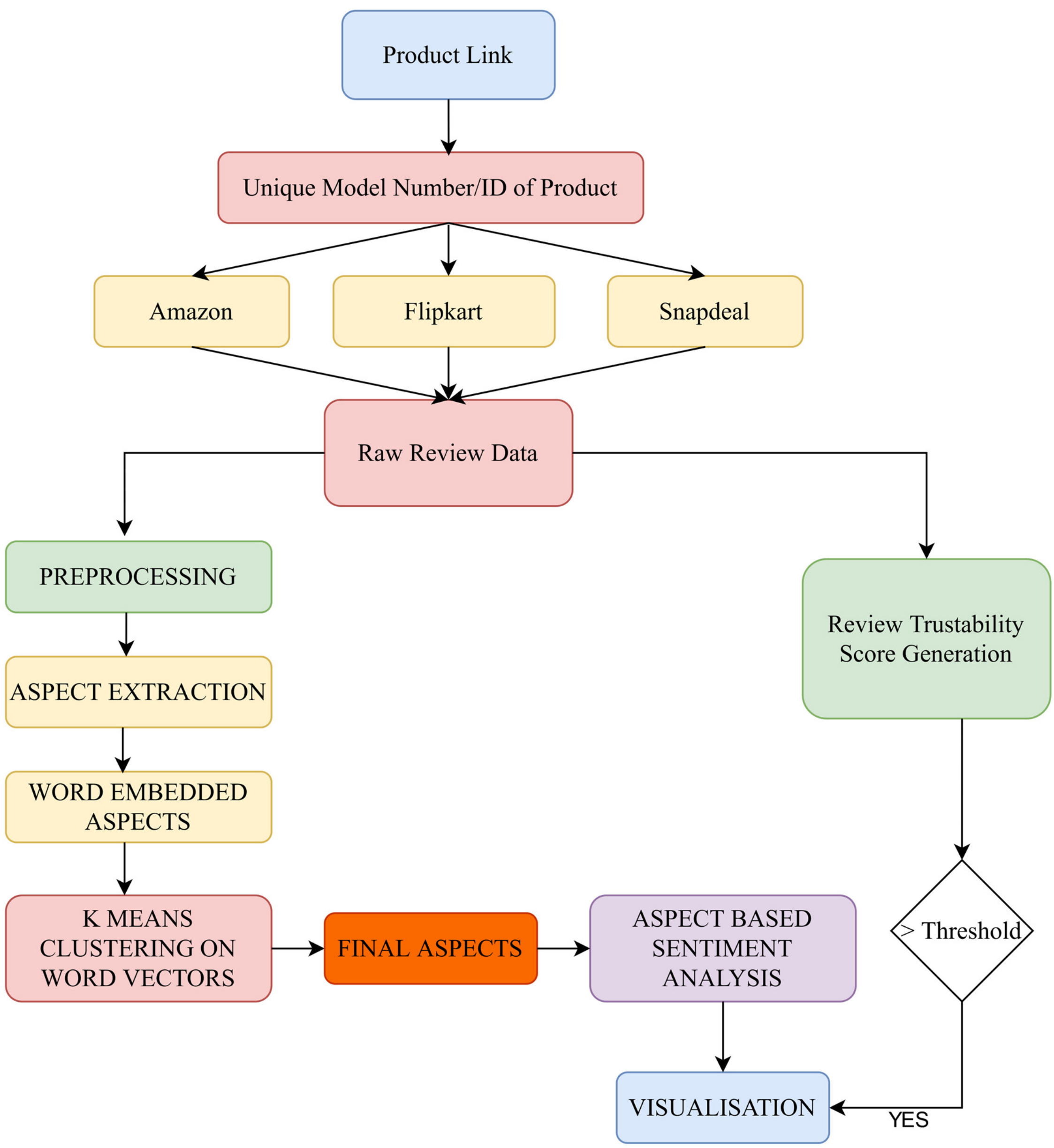

Figure 2: System architecture of multi-domain ABSA.

3.1 Dynamic multi-domain review scraping

The Python package Selenium is used to carry out the review scraping procedure. This package is useful for automated browsing. Selenium enables web page interaction, website browsing, and HTML code parsing. The technique uses the product's particular model number to look for it on numerous websites. The following three steps are involved.

3.1.1 Individual ID recognition

The system launches a browser and uses Selenium to visit the link from the product's link. Using the HTML tag ID from the webpage, the product's distinctive model number is derived. Multiple domains search for the product using the model number. It currently searches Snapdeal and Flipkart.

3.1.2 Finding the product

The product is found by parsing the HTML code of the search result pages and using the links to the appropriate products. The links to the recognized products' review pages are processed.

3.1.3 Scraping and compiling reviews

On the relevant review sites, Beautiful Soup uses tag IDs to identify the reviews, and then it extracts the review's text and other information. The data are saved to a .csv file and kept in a Pandas DataFrame. As a result, we automatically compile product reviews from various sources. The following characteristics are present in each review: title, rating, description, and upvotes for the review.

3.2 Review trustability and keyword extraction

The review trustability and keyword extraction stages are made to find credible reviews and, as a result, to extract the characteristics of the product under consideration. All stages of the process are included in the extraction stage.

3.2.1 Review trustability score

The trustworthiness score of a review denotes the degree to which the viewpoint may be relied upon. The four main components of a review that make up the score are as follows:

i. Length of Sentences – Longer evaluations are frequently more thorough and accurately describe the product. Effectively counted the sentences in each review using the spaCy program.
ii. Readability – Look at the words used, the sentence structures, and other elements to determine readability. It describes how straightforward the text is to read. Reviews that are simpler to read are frequently thought to be more trustworthy. Using the Automated Readability Index technique, we determined the readability score for a particular review. This approach takes word structure into account and is often based on a word list's percentage of simple words or the average number of syllables per word.
iii. Target words – When looking at a review, a reader focuses mostly on specific words to identify the review's structure. Each target word has been counted for how many times it appears in each review while also accounting for its benefits, drawbacks, pros, and disadvantages.

iv. The number of votes: Obviously, the reviews with the most votes are the most helpful.

The sum of the points for each component is used to calculate the review's overall trustworthiness score. The scores have been changed so that they range from [0,1]. Therefore, using the final scores of each review to determine whether a score greater than the threshold (average of normalized scores) applies, we arrive at a conclusion that is communicated at the visualization step.

The score of credibility is defined by considering the textual and sentimental factors of the reviews as follows:

$$SC = N(k_i = 0\ W_i * V_i), \tag{1}$$

where $N(x)$ stands for the normalization function, resulting in a [0,1] range, V_i reflects the value of a certain aspect, and W_i denotes the weighting given to various factors based on typical customer behavior.

In the proposed methodology, four factors are defined and weighted, as shown in Table 1.

Table 1: Textual and sentiment factors with weightages

S. no.	V_i	W_i
1	Sentence count	0.2
2	Reading comprehension	0.3
3	Target words	0.2
4	Upvoted votes	0.3

3.2.2 Pre-processing

The review content that has been scraped from numerous different domains is cleaned up by the cleaning software we have created. It gets rid of additional characters, hyperlinks, symbols, spaces, and other text patterns that our algorithms could not handle. We also replaced several period symbols with a single period. We extracted the text description for our study from this cleaned dataset. The other irregularities that predominate in the scraped evaluations are the data type and data organization within their connected attributes. We have tools for both typecasting the data into the format that is absolutely necessary and for removing the extraneous information that is included with the pertinent data.

3.2.3 Extraction of noun–adjective pairs

The first step here is to change the votes information into numeric representation in order to do make better study of attributes. The primary goal of this stage is to extract the attributes and any related modifiers (adjectives). We created a dependency parse tree using the Python tool spaCy in order to extract aspect-adjective pairings based on specific syntactic dependency paths. The result of this stage is a dictionary of these noun–adjectives, which is utilized as an input in the next step of grouping aspects (Figure 3).

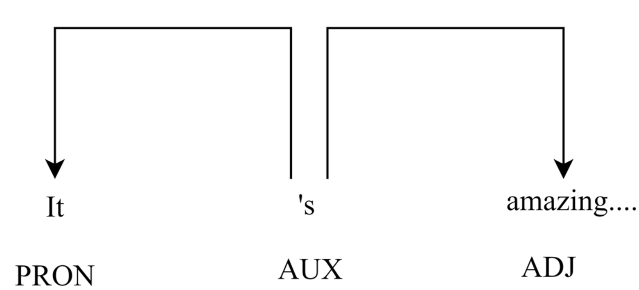

Figure 3: Dependency parse tree.

The rules were constructed from the terms in the parts-of-speech tags of the review text. For example, the phrase's noun would be a word with the "nsubj" dependence relationship to the verb token, and the noun's adjective would be a word with the "acomp" dependence relationship. In addition, we replaced every usage of the pronoun "product" with the word "Product," because, in its general sense, the pronoun refers to the complete product. As a result, this pair would be extracted as a relevant aspect–modifier pair. In the picture below, "A" stands for the aspect, while "M" and "M′" are the appropriate modifiers of the adjective for the aspect. This illustrates some of the ideas we came up with (Figure 4).

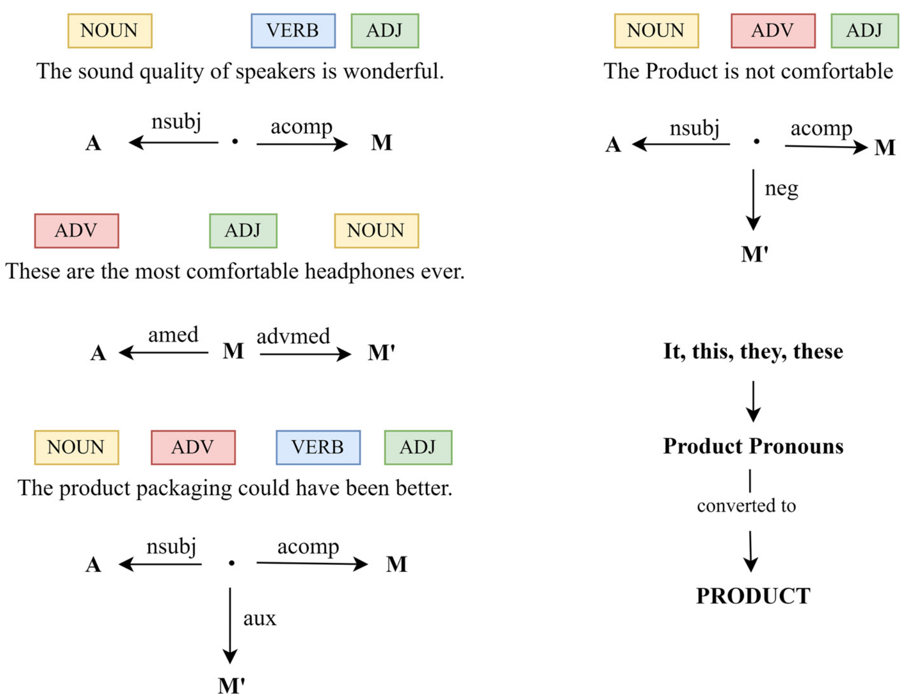

Figure 4: Rules explanation.

3.2.4 Grouping aspects

This is done in two steps. First step is aspect generation using word embedding process and the second step is clustering the generated aspects.

3.2.4.1 Word embedded aspects generation

We may detect how similar two words are by comparing the word vectors produced by the integrated high-performance vectorization model of spaCy. Using spaCy for vectorization allows you rapid and simple access to over a million different word vectors, and unlike other libraries like NLTK, its multi-task convolution neural network model is trained on "web" data rather than "newspaper" data.

3.2.4.2 Clustering aspects

SciKit Learn's *K*-Means technique is then used to group the word vectors. We obtained effective results for 15 clusters using the *K*-Means approach, which varies in effectiveness depending on the quantity of data provided. The term that most often appeared in each cluster was used to designate the clusters.

3.2.5 Determining polarity scores

To identify the polarity of the aspect's adjectives, we utilized the NLTK library's Vader Sentiment Analysis approach. We chose this method over the spaCy and Text Blob approaches by taking speed and accuracy into consideration. We calculated the compound polarities of the set of derived adjectives and then added them all together to get the final aspect polarity.

3.2.6 Visualization

We utilized the Matplotlib library to display the ABSA results and the review's credibility. In contrast, the ABSA bar plots display the final most popular subjects across various domains, such as Flipkart, Amazon, and Snapdeal, as positive and negative bars with appropriate polarity values. Review trustability bar plots give information about the demographics and reviews that were assessed.

4 Results and discussions

The technology we built can dynamically extract reviews from many domains. Currently registered domains include Flipkart, Amazon, and Snapdeal. For the majority of the products, there are several reviews available on the most well-known websites, Amazon and Flipkart. Snapdeal continued to create fewer reviews. Getting a specific model number from Flipkart is the first step in the process, which is then utilized to obtain information from Amazon and Snapdeal. After scraping the reviews using the techniques, the keywords of a text are retrieved along with the corresponding adjectives, and the polarity for these qualities is established along with the trustworthiness score for each review. The keywords or elements provide a succinct but impactful grasp of the material by summarizing the information in the text. The example on the "Dell Inspiron Laptop" is discussed further.

4.1 Elements

The system looks up the goods on Amazon, Flipkart, and Snapdeal and displays bar graphs of the analyzed data. Let us examine each experiment performed on Amazon, Flipkart, and Snapdeal for the analysis of product reviews using our method and then compare them all.

The 15 aspects of the product that have received the most attention on Amazon are shown in Figure 5. We can see that the abundance of information is allowing our system to effectively capture many of the reviews posted on Amazon. A few elements have negative and neutral attitudes, while most of the aspects have good opinions.

The 15 aspects of the product that have received the most attention on Flipkart are shown in Figure 6. The system effectively identified the topics that were discussed and showed varied behavior among the topics, showing opinions that were both favorable and negative.

The five components of the product that have received the greatest attention on Snapdeal are shown in Figure 7. The analysis's capacity to present various features is impacted by the information's restricted availability, as can be seen in Figures 5 and 6. The well-known cold start issue occurs as a result of insufficient data availability. A user is given confusing reviews and ratings because, unlike Amazon and Flipkart, the elements are largely good. Our suggested system, which takes into account the information from all three of the aforementioned domains and conducts an overall analysis, resolves this.

The 15 product features that have received the most attention on Amazon, Flipkart, and Snapdeal are represented in Figure 8. In contrast to current techniques, the characteristics are grouped based on the

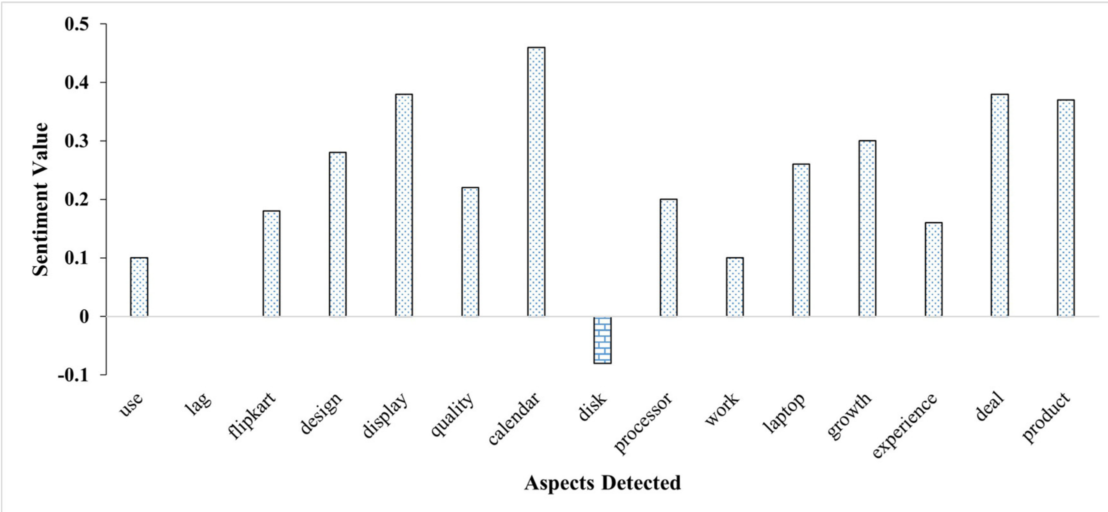

Figure 5: Amazon's aspects with corresponding sentiments for "Dell Inspiron Laptop."

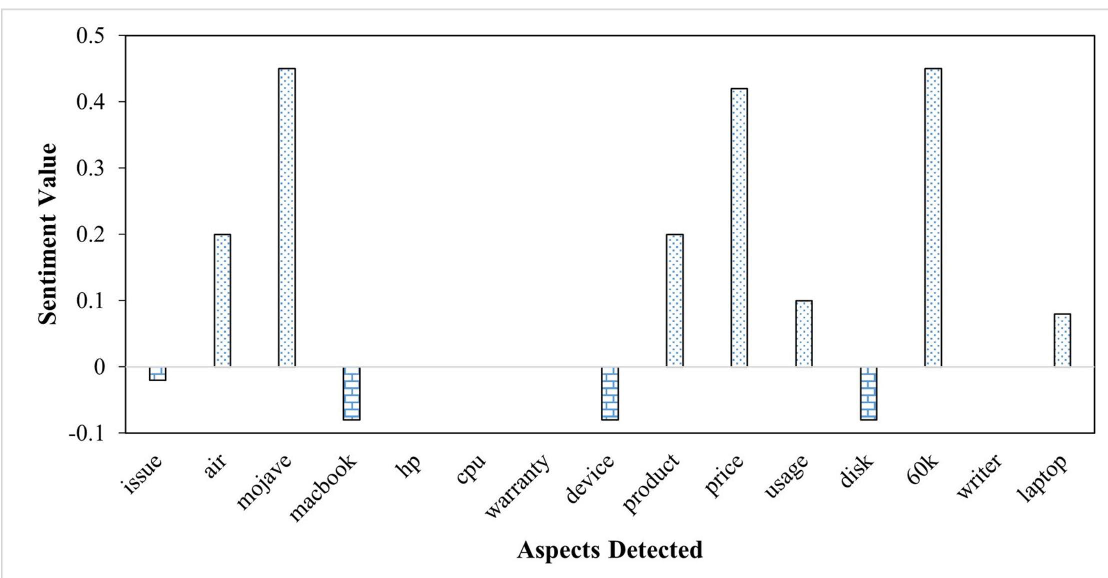

Figure 6: Flipkart's aspects with corresponding sentiments for Dell Inspiron Laptop.

semantics of the aspects. The most frequent characteristic within each cluster also emerges as the cluster's final element. In this manner, 15 clusters of various viewpoints produce 15 distinct features that effectively depict the product. This avoids the prejudice of only showing the top 15 often-mentioned features overall and instead displays all pertinent aspects of all the perspectives discussed. By expanding the richness of the aspects and the needs of the consumers, this analysis surpasses all other studies.

The experiments conducted on individual and integral data sources depict the performance of the proposed model w.r.t. the determined aspects and the corresponding sentiment of the particular aspect. For example, the aspect named "user" in Figure 7 is more inclined to positive polarity around 0.6 due to the lower volume of reviews, whereas in Figure 8, it is showing less than 0.2. In addition, a few aspects that exhibit negative sentiment polarity were not actually considerable when the volume of the input increased. This infers that the dynamics in the weightage of the aspects in the reviews improved with a high distribution and less bias.

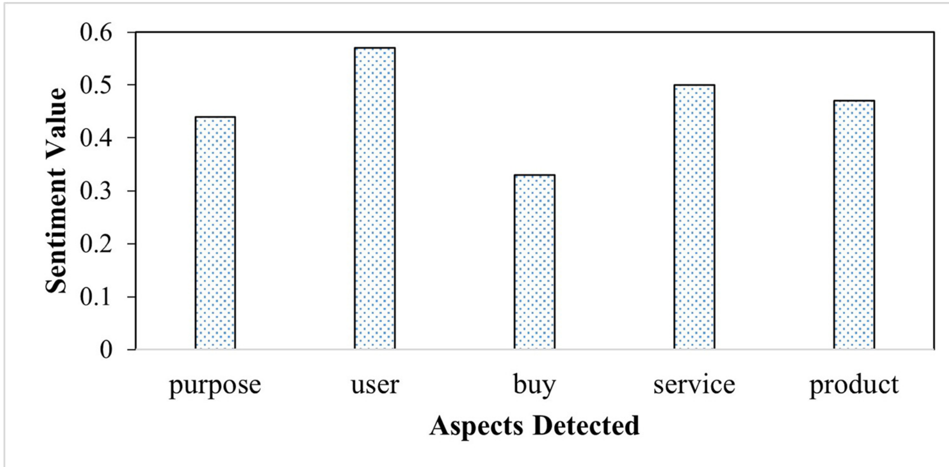

Figure 7: Snapdeal's aspects with corresponding sentiments for Dell Inspiron Laptop.

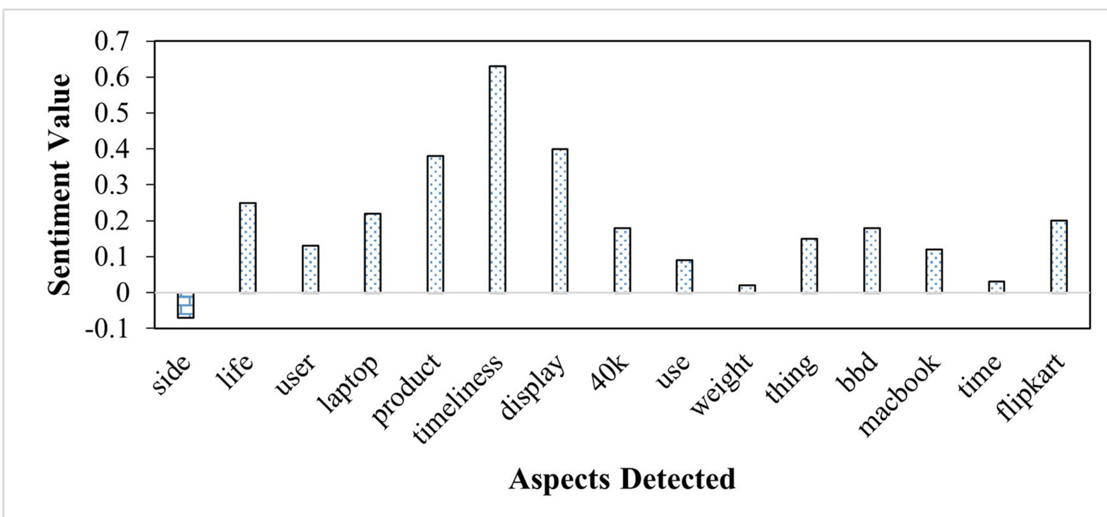

Figure 8: Dynamic multi-domain aspects with corresponding sentiments.

4.2 Review trustability

One important issue resolved in our research was the reliability of reviews. The important elements of sentence length, readability of review content, amount of upvotes, and target terms have been used to assign a trustability score to each review that is scraped from various domains. We avoid any prejudice that might result from showing the most helpful review based merely on the number of upvotes by taking into account all of these indicators.

The majority of e-commerce companies simply use the number of positive reviews as a quantitative criterion for customers to determine how trustworthy a review is. Figure 9 illustrates how our algorithm evaluates the reliability of reviews using four key metrics (Votes, Sentences, Triggering Words, and Readability). By addressing biases in the current system, this technique can direct customers to reviews that are beneficial. The average of the normalized scores generated yields a threshold.

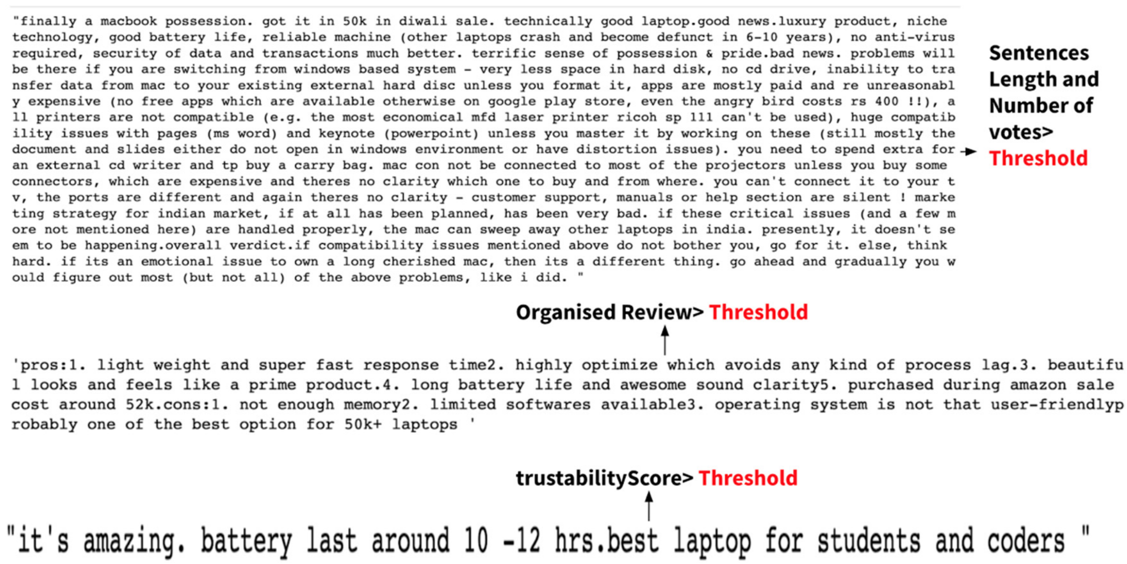

Figure 9: Example of reviews recognized as per trustability score threshold.

As a result, it can be said that the system created provides the client with relevant information that he might not discover while browsing an e-commerce website and is, for the most part, an improvement over the review presentation method used by those websites (Table 2).

Table 2: Bottom-most reviews based on the trustability score

Comment	Trustability
ok	0.00
go for it	0.056
i love it	0.066
wow	0.075
good for 50k	0.162

Customers do not respect reviews that are just a description of a product, as seen in Figure 9, and as a result, the trustworthiness score is lower. Organizations can greatly benefit from this because it expands the field of use for feedback analysis.

5 Conclusion

The proposed work will make a substantial contribution to the removal of bias from the current commercial system. The review corpus expands when more domains are used as extraction sources for a single product. As a result, the product is represented more accurately, and customers have less trouble using different apps. Furthermore, it helps reveal prejudice that is specific to websites. The approach works well for products with a certain model number or ID that are correctly supplied. As a result, the system uses dynamic multi-domain scraping to overcome the cold start problem. The availability of data is one of the scraping's limitations. The scrape is limited to those things solely because the model number is identified for many significant and tangible goods.

In the phase of AE, noun–adjective pairs are identified by the use of natural language processing techniques. The most effective method of detecting features is through this type of grammatical analysis. The information from many aspects was combined into a much smaller number of aspect clusters once the aspects were clustered. The adjectives in the questions rely on their context, which is important to keep in mind. The polarity values that are generated, which are more accurate than those from context-free sentiment analysis, are built on these context-based adjectives.

Acknowledgments: It would be our pleasure to extend our sincere declaration to the Department of Information Technology at Chaitanya Bharathi Institute of Technology in Hyderabad, India for continuous support in carrying this research.

Author contributions: Mukkamula Venu Gopalachari has conceptualised the proposed approach, executed the architecture, performed the experiments, collected, and concluded the results. Sangeeta Gupta and Salakapuri Rakesh has validated the concept of approach, analysed the methodology and results. Dharmana Jayaram and Pulipati Venkateswara Rao audited the whole architecture of the approach and experimental results.

Conflict of interest: Authors state no conflict of interest to disclose.

Data availability statement: The dataset used for this research is available online and has a proper citation within the article's contents.

References

[1] Ananthajothi K, Karthikayani K, Prabha R. Explicit and implicit oriented aspect-based sentiment analysis with optimal feature selection and deep learning for demonetization in India. Data Knowl Eng. 2022;142:102092.
[2] Alyami S, Alhothali A, Jamal A. Systematic literature review of Arabic aspect-based sentiment analysis. J King Saud Univ - Comput Inf Sci. 2022;34(9):6524–51.
[3] Araque O, Corcuera-Platas I, Sánchez-Rada J, Iglesias C. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. Expert Syst Appl. 2017;77(19):236–46.
[4] Lu Q, Sun X, Sutcliffe R, Xing Y, Zhang H. Sentiment interaction and multi-graph perception with graph convolutional networks for aspect-based sentiment analysis. Knowl Syst. 2022;256:109840.
[5] Dai X, Bikdash M, Meyer B. From social media to public health surveillance: Word embedding based clustering method for Twitter classification. SoutheastCon 2017; 2017. p. 1–7.
[6] Khan M, Alam M, Basheer S, Ansari MD, Kumar N. A map reduce clustering approach for sentiment analysis using big data. Cognit Sci Technol. 2022;1:223–9. doi: 10.1007/978-981-19-2350-0_22.
[7] Venugopalan M, Gupta D. An enhanced guided LDA model augmented with BERT based semantic strength for aspect term extraction in sentiment analysis. Knowl Syst. 2022;246:108668.
[8] Kamkarhaghighi M, Makrehchi M. Content tree word embedding for document representation. Expert Syst Appl. 2017;90:241–9.
[9] Wang W, Pan SJ, Dahlmeier D, Xiao X. Recursive neural conditional random fields for aspect-based sentiment analysis. Proc. Conf. Empirical Methods Natural Lang. Process; 2016. p. 616–26.
[10] Ma Y, Peng H, Khan T, Cambria E, Hussain A. Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis. Cogn Comput. 2018;10(4):639–50.
[11] Luo H, Li T, Liu B, Wang B, Unger H. Improving aspect term extraction with bidirectional dependency tree representation. IEEE/ACM Trans Audio Speech Lang Process. 2019;27(7):1201–12.
[12] Pontiki M, Galanis D, Papageorgiou H, Manandhar S, Androutsopoulos I. Semeval-2015 task 12: Aspect based sentiment analysis. Proc. 9th Int. Workshop Semantic Evaluation; 2015. p. 486–95.
[13] Yang J, Yang R, Wang C, Xie J. Multi-entity aspect-based sentiment analysis with context, entity and aspect memory. Proceedings of 32nd AAAI Conference on Artificial Intelligence; 2018. p. 6029–36.
[14] Wang J, Li J, Li S, Kang Y, Zhang M, Si L, et al. Aspect sentiment classification with both word level and clause-level attention networks. Proceedings 27th International Joint Conference of Artificial Intelligence; 2018. p. 4439–45.

[15] Angelidis S, Lapata M. Multiple instance learning networks for fine-grained sentiment analysis. Trans Assoc Comput Linguist. 2018;6:17–31.
[16] Chi CGQ, Ouyang Z, Xu X. Changing perceptions and reasoning process: Comparison of residents' pre-and post-event attitudes. Ann Tour Res. 2018;70:39–53.
[17] Huang H, Zhang B, Jing L, Fu X, Chen X, Shi J. Logic tensor network with massive, learned knowledge for aspect-based sentiment analysis. Knowl Syst. 2022;257:109943.
[18] Yang L, Na J-C, Yu J. Cross-modal multitask transformer for end-to-end multimodal aspect-based sentiment analysis. Inf Process Manag. 2022;59(5):103038.

mathematics MDPI

Article

A Novel Approach to Integrating Uncertainty into a Push Re-Label Network Flow Algorithm for Pit Optimization

Devendra Joshi [1], Marwan Ali Albahar [2], Premkumar Chithaluru [3,4,*], Aman Singh [5,6], Arvind Yadav [1] and Yini Miro [5,7]

1 Department of CSE, Koneru Lakshmaiah Education Foundation, Guntur 522002, India
2 Department of Computer Science, Umm Al Qura University, Mecca 24382, Saudi Arabia
3 Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Hyderabad 500075, India
4 Department of Project Management, Universidad Internacional Iberoamericana, Campeche C.P. 24560, Mexico
5 Higher Polytechnic School, Universidad Europea del Atlántico, C/Isabel Torres 21, 39011 Santander, Spain
6 Uttaranchal Institute of Technology, Uttaranchal University, Dehradun 248007, India
7 Department of Engineering, Universidad Internacional Iberoamericana, Arecibo, PR 00613, USA
* Correspondence: bharathkumar30@gmail.com

Abstract: The standard optimization of open-pit mine design and production scheduling, which is impacted by a variety of factors, is an essential part of mining activities. The metal uncertainty, which is connected to supply uncertainty, is a crucial component in optimization. To address uncertainties regarding the economic value of mining blocks and the general problem of mine design optimization, a minimum-cut network flow algorithm is employed to give the optimal ultimate pit limits and pushback designs under uncertainty. A structure that is computationally effective and can manage the joint presentation and treatment of the economic values of mining blocks under various circumstances is created by the push re-label minimum-cut technique. In this study, the algorithm is put to the test using a copper deposit and shows similarities to other stochastic optimizers for mine planning that have already been created. Higher possibilities of reaching predicted production targets are created by the algorithm's earlier selection of more certain blocks with blocks of high value. Results show that, in comparison to a conventional approach using the same algorithm, the cumulative metal output is larger when the uncertainty in the metal content is taken into consideration. There is also an additional 10% gain in net present value.

Keywords: open-pit mine; ultimate pit limit; uncertainty modeling; minimum cut; network flow

MSC: 68W40; 68W50

Citation: Joshi, D.; Ali Albahar, M.; Chithaluru, P.; Singh, A.; Yadav, A.; Miro, Y. A Novel Approach to Integrating Uncertainty into a Push Re-Label Network Flow Algorithm for Pit Optimization. *Mathematics* **2022**, *10*, 4803. https://doi.org/10.3390/math10244803

Academic Editors: Ioannis E. Livieris and Ioannis G. Tsoulos

Received: 3 November 2022
Accepted: 12 December 2022
Published: 16 December 2022

1. Introduction

A challenging issue in mine planning and design is open-pit mine production scheduling. To maximize the total discounted profit, an open-pit optimization problem is used to remove mining blocks from the pit while satisfying all of its constraints. Reserve constraints: the restriction that a block can only be mined once during its lifetime. Slope constraints: a block cannot be mined before its predecessors. A group of overlying blocks must be removed to reach a given block. Mining constraints: to ensure the effective use of mining equipment, the total weights of blocks mined during each period should be at least equal to a minimum mining limit. On the other hand, it should not be greater than the capacity of the mining equipment that was in use at the time. Both the upper bound and the lower bound must be satisfied under mining constraints. Processing constraints: this primarily depends on the processing capacity of the plants. The total number of ore blocks mined during each period should at least be equal to the minimum number needed for processing, but it should not exceed the processing plant's capacity because the excess ore must then be stored somewhere. Metal production constraints: Both the upper and lower bounds of the metal production limitations must be satisfied, and the amount of metal recovered from the ore blocks processed should not be less than a minimum amount and should not exceed the amount that can be sold during this period. Open-pit optimization is typically defined as a type of integer or mixed integer programming issue that can be solved utilizing a commercial solver. A three-dimensional array of blocks is used to depict the orebody. With the limited number of exploration drilling data and the use of geostatistical algorithms, the mineral grade of each block within the orebody is determined. The volume, grade, and tonnage details are provided for each block. It is well recognized that orebody models and their geological features are a significant source of scheduling risk. Every block has a block economic value that represents the corresponding net profit. The challenge of allocating a mining block's extraction sequence to various production periods is the main focus of open-pit mine design. A mining block's weight, block economic value, ore contents, metal contents, and other characteristics are used to describe it. Many mining blocks are assigned to distinct production periods to maximize profit within predetermined bounds. These restrictions not only include, but also contain: (a) a block must be allocated to one production period; (b) a set of neighboring blocks for each block must have the same production period or prior periods; and (c) a sufficient quantity of blocks must be assigned during a specified time frame to ensure that the total quantity of material, ore, and metal recovered from the mining falls within the allowable limit for a particular production period. An open-pit mine's production schedule is a challenging issue [1–3]. Usually, mixed integer programming is used to resolve open-pit production scheduling [4]. Finding the best production schedule is a challenging task because of the huge number of integer variables, numerous limitations, and lack of certainty around the various mining block parameters. The main source of uncertainty is geological, which implies that the metal composition of a block has been accurately determined previously. Additionally, the economic fluctuation of metal prices may cause uncertainty [5]. Numerous optimization techniques have been suggested to solve the deterministic production scheduling while treating the mining block values as known and ignoring their uncertainty. Mixed-integer linear programming methods have been proposed to optimize the net present value for many mining projects [6,7]. An alternative approach built on Lagrangian relaxation was suggested by Dagdelen and Johnson [1]. For determining the production plan of an open-pit mine, Caccetta and Hill suggested a branch-and-cut approach [8]. The Fundamental Tree Algorithm (FTA), created by Ramazan [9], decreases the size of the real problem by aggregating mining blocks. To address the mine production scheduling issue, Bley et al. [10] suggested a technique based on the cutting-plane method. For enhancing the production scheduling problem's computational effectiveness, Bienstock and Zuckerberg [11] presented a heuristic method. A mine planning framework based on network flow is presented by Topal and Ramazan [12], who also illustrate a significant application in a real mine. Using a heuristic-based linear relaxation, Chicoisne et al. [13] tackled a bigger size (3 million blocks) mine process optimization issue. A solution based on sliding windows has been suggested to address deterministic mine design [14,15]. Mining block sequencing issues are addressed by Lambert and Newman [16] using a Lagrangian relaxation-based strategy. Lamhgari et al. [17] use metaheuristic techniques to handle production scheduling in open-pit mines, such as variable neighborhood descent. It takes a lot of effort to incorporate mining waste management and removal into the MILP optimization of the project plan [18]. The mining sector has been using deterministic models for open-pit optimization since the 1980s [19–23]. Unfortunately, the basic presumption that grade and quantity will always be constant in a given block oversimplifies the issue and results in an inaccurate evaluation [24–27]. The block grades, and subsequently the metal contents, are computed using a finite number of samples, resulting in usual uncertainties with these usually expected values [28]. Numerous simulated orebody models are created using geostatistical modeling techniques to account for the geological uncertainty regarding block grades and metal concentration [28–31]. Therefore, when including these uncertainties, the stochastic approach is more accurate. By maximizing the net present value and reducing the variance from the target, some authors proposed uncertainty-based approaches based on the concept of geological risk discounting [15,32]. A multi-stage stochastic programming approach was investigated by Boland et al. [33] to handle different aspects of mine production scheduling.

However, Godoy and Dimitrakopoulos [25] suggested a different strategy that would utilize a simulated annealing process to minimize the computing time. To optimize a mine schedule, researchers used a series of pit shells created by the layered application of the Lerchs and Grossmann approach [34–36]. They have demonstrated that their techniques are computationally faster than the conventional method, boosting the project value by 15% to 28% [34,35]. For resolving the computing challenges related to full-scale stochastic integer programming, Lamghari and Dimitrakopoulos [17] suggested a metaheuristic approach and created an algorithm combining Tabu search. For stochastic production planning, a two-stage stochastic integer programming method with a stockpiling choice and geological risk discounting is proposed [37]. The preceding model is expanded to include mining complexes [27,38]. To handle production schedules as well as waste dumping, Rimélé et al. [39] adopt a two-stage stochastic integer programming method.

In this study, the push re-label minimum-cut method is improved and tested in terms of the design of pushbacks and optimal pit bounds under uncertainty. The latter is produced by parameterizing the minimum-cut graph's arc capacities. Similar to the commercial Lerchs and Grossmann algorithm implementations, the time value of money and the associated discounting of block economic values are implemented indirectly and are based on bench-wise scheduling and predefined mining capacity. A test case on a copper mine serves to illustrate the procedure's complexity. It should be noted that the case study maintains generality by using numerous economic values for mining blocks that were produced from simulated realizations of the resource. Combining simulated orebody models, price and exchange rate forecasts cost forecasts, and other forecasts could produce economic values. The method given here is compared against the comparable deterministic variant, using only deterministic inputs, in this comparative research. The push re-label minimum-cut algorithm is initially described in the following sections about mine design and optimization. The push re-label minimum-cut approach is described after that in a brief explanation. Some similarities to the deterministic scenario are presented.

The paper continues with a literature survey. The solution approach is discussed in the part titled "Methodology," and its application to a copper deposit is covered in the section titled "3.1. A minimum-cut graph to optimize ultimate pit limits, 3.2. Uncertainty implementation in a minimum-cut graph, 3.3. Pushback design using arc capacity parameterization in a minimum-cut graph, 3.4. A mathematical formulation of a stochastic mine pit optimization under uncertainty." The paper is finished with the sections "Results" and "Conclusion and future scope."

2. Literature Survey

According to the number of highly possible scenarios, metal and geological uncertainties are treated by utilizing a spatial stochastic modeling approach to mine maps of grade, metal content, and geology. The spatial correlation among observations from the drilling procedure and the volumetric variances between the available data and the supposed mining blocks are explicitly taken into account by these algorithms [40–42]. By increasing the extraction duration of the high-risk blocks, the authors created the hypothesis of orebody risk discounting, which postpones the risk for future timeframes [43]. The main issue with the abovementioned considerations is that they fail to take into account simulation results, a risk that has been simultaneously analyzed as sets of blocks in mine from period to period, and previously given risk probability for each block [43,44]. Ramazan and Dimitrakopoulos provided the first mathematical models that successfully depicted the stochastic characteristics of mine design [45]. They investigate a two-stage stochastic model with multiple equivalent possibilities to represent unpredictable geology. Each scenario has two parts: the first phase addresses the mining sequence, and the second phase addresses any variation from the target. These two phases make up the decision-making mechanism. The goal is to increase the expected net present value as much as possible while minimizing production target deviations, which reduces the risk that production targets will not be met.

The idea of simulated annealing was initially developed [25] and further investigated [34,46] for long-term production schedules in a stochastic approach. This approach was enhanced to take into account various mines, inventories, and operational sites [27]. A stochastic integer linear programming approach was created by Ramazan and Dimitrakopoulos [37] to optimize the net present value while reducing the deviation from production goals. A test case using the stochastic approach under risk was reported by Chatterjee and Dimitrakopoulos [46]. This makes it more challenging, but also more advantageous, to include geological risk in the optimization method. These advantages were first emphasized [25,34,37,46–48].

Geostatistical techniques can be used to measure, model, and quantify uncertainty. This is accomplished by utilizing any geostatistical simulation technique to generate numerous equal probability scenarios of orebody realization. Risk can be minimized by incorporating uncertainty into decision-making processes. As a result, the team responsible for mine design and production schedules will be able to achieve higher profit margins and develop a more effective risk management plan. The mining sector, on the other hand, is well conscious of the uncertainties and risks. Stochastic models receive significant attention since they manage uncertainty and risk in a way that is more practical and realistic. Evaluating the revenue under different scenarios and attempting to reduce production target deviations is a successful strategy for resolving stochastic problems [34,46,48].

Since geological uncertainty directly affects the supply of ore and metals, mining industries believe that it is the main cause of their unrealized cash flow projections. The challenge concerning how to determine the supply of ore for processing arises. It is a complicated issue to answer since it depends on the removal sequence throughout time in addition to the ore's spatial variability The same resource will yield various ore supplies depending on the extraction methods used. Because it depends on economic factors and changes over time, the definition of ore changes over time. The concept of accessible ore availability has typically been assessed under the presumption of constant technical and financial limitations in both space and time. Throughout the traditional mine production and design framework, the aggregate form of ore grade assumption is utilized in combination with geological, financial, and ecological limitations to create the extraction schedule that yields the highest economic benefit. The utilization of risk-free models revealed a considerable gap between prediction and real financial results.

Several studies have investigated the effect of geological uncertainty on project economics and employed conditional simulations to undertake a risk analysis of mine plan technical specifications [24,49]. The existence of risk modeling techniques facilitates the development of novel scheduling systems that incorporate simulated geological uncertainties in mine strategic planning. The initial pushback design's stochastic graph closure problem is a relaxed scenario in which resource limitations are not considered. Resource limits must be applied to create pushbacks. Several techniques are addressed in the various literature. The suggested algorithm's major benefit is that it can be computationally very quick; therefore, incorporating uncertainty into production planning is regularly viable.

3. Methodology

The proposed approach is conceptually presented as (i) a minimum-cut graph to optimize ultimate pit limits.

(ii) Uncertainty implementation in a minimum-cut graph.

(iii) Pushback design using arc capacity parameterization in a minimum-cut graph.

3.1. A Minimum-Cut Graph to Optimize Ultimate Pit Limits

A simple, two-dimensional example can be used to illustrate open-pit optimization as a minimum graph cut problem. Figure 1 depicts a section where the associated economic value is represented by the numerical values inside each block. Based on the expected grade, metal selling price, mining cost, and processing cost, a block's economic worth is determined. Blocks with positive values are ore blocks, while blocks with negative values are waste blocks. Figure 2 illustrates a directed graph as a possible representation of this two-dimensional orebody model. The directed graph is thought of as having each mining block as a node. A source and sink are two unique nodes. The model's ore block nodes are linked to the source node, and the capacity of those arcs represents the economic value of blocks. However, every node within the model that is defined as a waste block is linked to a sink node, and the capacities of those arcs are equal to the absolute values of waste block nodes. One needs to gain access to that specific block to mine a certain block, $x(i, j)$. It is impossible to reach a specific block and mine it unless the blocks above it are removed; this is made feasible by retaining slope limits. To meet slope limitations, it is required to identify the underlying blocks that need to be eliminated before removing the desired block. The red arrow in Figure 2 indicates a block's slope constraints. The slope constraint arcs' capacities are given an infinite value. Slope constraint arcs can never be in the minimum cut since their value is indefinite. As a result, the minimum cut will result in an acceptable pit because the slope constraint will not be satisfied. The optimal ultimate pit can be found by transforming the example of a two-dimensional orebody into a directed graph problem and then applying the minimum-cut method. A set of directed arcs with at least one arc in each direction from the source to the sink node forms a cut of a directed graph. There will not be a straight path between the source and sink nodes if the arcs in the cut are eliminated. The cut value is the total of all of the arcs in the cut's flow capacity in the source-to-sink direction.

2	5	-2	2	-2
-3	6	-1	4	-2
-5	8	4	-7	-4

Figure 1. Block model with block economic values.

Finding the cut in the graph where the total of the capacities is least across all cuts is the goal of the minimum-cut problem. Figure 2 shows the minimum cut of the two-dimensional open pit that was indicated in Figure 1 with a bold, brown line. The orebody model in this example has a minimum cut value of 8. In this illustration, the ore blocks are on the sink node of the cut, and the waste blocks are on the source node. The minimum cut in the open-pit optimization issue is the cut that minimizes the sum of the number of waste blocks on the source node side and the sum of the number of ore blocks on the sink node side. In other words, the minimum cut maximizes the number of ore blocks present in the pit while simultaneously reducing the waste blocks present in the pit. Given that the slope restrictions are followed, the minimum cut is likewise a valid pit. When the capacities of the ore blocks on the sink side and the waste blocks on the source side are added together, it was determined that the cut shown has value (3 + 1 + 2 + 2) = 8, which is the same as this problem's minimum cut value.

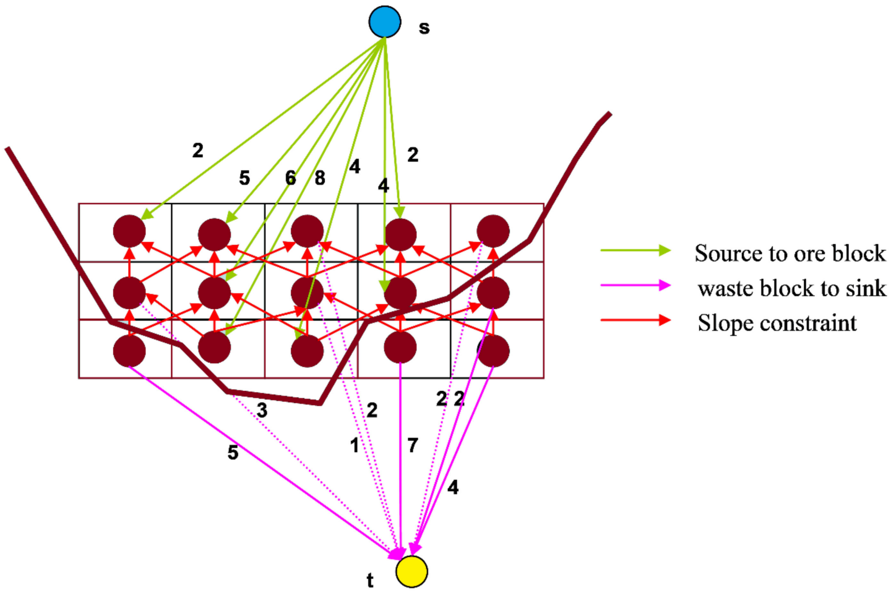

Figure 2. Pit limit calculation using the minimum directed graph cut.

3.2. Uncertainty Implementation in a Minimum-Cut Graph

Multiple orebody models can be simply added to the previously described technique. Figure 3 illustrates the same directed graph that is produced utilizing numerous orebodies, as in the previous example. All ore blocks will be linked to the same source node, and all waste blocks will be linked to the sink node. Using arcs with infinite capacity, the slope constraint will be maintained. A specific block in simulated orebody models may be wasted in one simulated block model and ore in another. Despite the computer simulation, the source node is linked to all ore blocks, while the sink node is connected to all waste blocks. This suggests that a certain block can be connected to the source node in one realization while also being connected to the waste node in another realization. The arc capacity will be different. Using the top-left block in Figure 3 as an example, let us say that the ores in Simulations 1 and 2 have economic values of 2 and 1, respectively. The source node is connected to that block in Simulations 1 and 2 with arc capabilities of 2 and 1, respectively. The same block, however, has a block economic value of -6 in Simulation 3, and an arc with an arc capacity of 6 is constructed from the block to the sink. Integrate the data from all simulations, and this is comparable to creating a directed graph with the number of nodes equal to n times the number of blocks in the deposit, where n is the number of simulated orebodies. The values of the blocks can be created using simulated grades, simulated grades paired with simulated commodity price forecasts, simulated grades combined with simulated commodity price forecasts and mining/processing expenses, etc. This allows for the integration of any uncertainty related to the estimation of a block's value.

Multiple orebody models can be used to make the directed graph, and then the minimum-cut technique can be used to create an ultimate pit. The identical block from the various simulations may not lie on the same side of the minimum cut if the minimum-cut approach is applied to the directed graph defined above with n no. of simulations. The main difficulty in formulating the minimum-cut technique for addressing the open-pit optimization problem with simulated orebodies is that, to produce an ultimate pit, a given block from various simulations must be on the minimum-cut side. No matter how many simulations are performed, the choice should be made in terms of a block as either being in the ultimate pit or not. To guarantee that a specific block lies on the same side of the cut for all simulations, another constraint must be added to the graph. It is possible to implement this constraint by combining blocks from many simulations into a bidirectional arc with infinite capacity. This ensures that there will never be a situation in which the same block from different simulations will fall on different sides of the minimum cut because these bidirectional arcs have infinite capacity and will never be in the minimum cut. So long as all requirements are respected, the pit produced by the minimum-cut algorithm with various orebody models is valid. These nodes can be combined into a single node because the same block will appear in the same simulation on the same side of the minimum cut. A single arc can be created by merging the arcs from the source node to the merged nodes from various simulations, and the arc's capacity will equal the total of all the capacities when that particular block is ore in all scenarios. Similar to the previous example, a single arc can be created from a merged node to the sink node, and its capacity will equal the sum of the actual capacities of the block's economic values in all simulations when the block is wasted. Figure 4 illustrates this idea. The graph there was created by integrating the three simulated orebody models from Figure 3. Each block in this figure has two values assigned to it. If the realizations are ore, the value in the top-left corner represents the total block economic values across all realizations. If any realizations are wasted, the value in the bottom-right corner represents the total relative block economic value. The value in the top-right corner represents the arc capacity from the source to that node, and the value in the bottom-left corner represents the arc capacity from the node to the sink. In Figure 4, for illustration, the top-left block's arc capacity from the source to a node is 3, which is the sum of 2 and 1, the block economic values of Simulations 1 and 2, respectively. With a capacity of 6, the absolute block value in Simulation 3, the same node is connected to the sink.

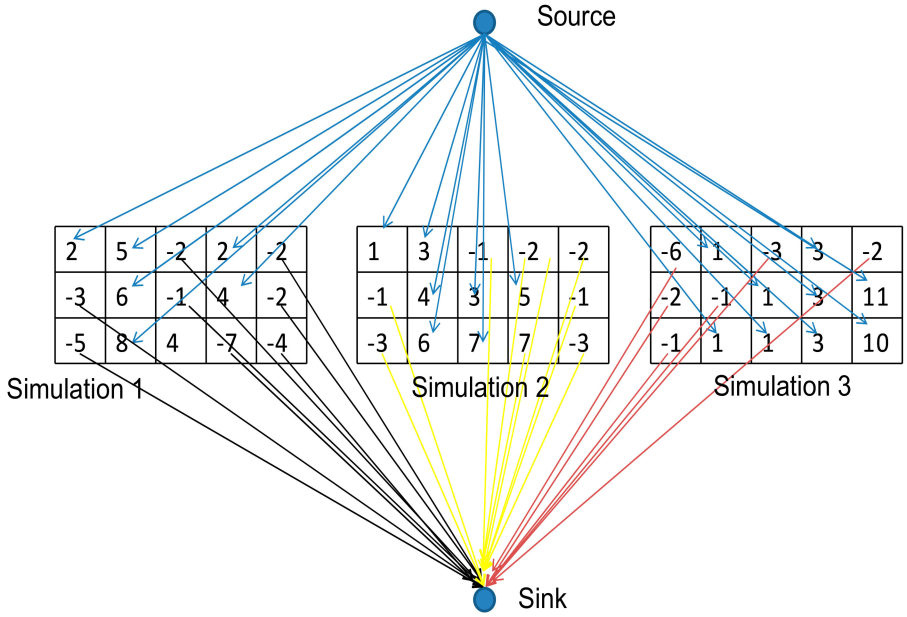

Figure 3. Multiple orebody models were used to create the graph.

The reduction in the number of nodes in the graph is the key benefit of the suggested strategy with different orebody models and economic values. Except for the source and sink nodes, the network has the same number of nodes as there are blocks. Due to this, regardless of the number of simulated orebodies, a deposit's node count will always be the same. The capacity of the arcs is the only thing that varies when the number of simulated models is altered. To take into consideration the geological risk in open-pit design, a stochastic variant of the network flow algorithm is utilized in the block economic value. To manage uncertainty, the stochastic network flow technique uses simulated models of an orebody that are equally probable. Calculations are performed using a set of scenarios rather than a precise and perhaps inaccurate model. This maximizes the net present value for an open-pit mine, given the uncertainty from limited data and orebody models. The method works as follows: the economic block value of each block in each simulation is determined when the simulations are run. The blocks with a negative block economic value are then connected to a sink that is the same for all simulations (one single sink node).

The same is done with blocks with positive block economic value; these are attached to a single source node and precedence restrictions must be followed. Furthermore, blocks with the same grid location (x, y, z) will have an endless number of capacity arcs connecting them. Because just one pit is being created, the additional constraint exists because the same blocks in the grid must be either within or outside the pit for each simulation. The algorithm then merges the blocks at the same place into a single block (node) with no more than one arc from the source node and one arc to the sink node, where these arcs have the capacity of the sum of the capacity of the arcs at the same place in the same simulation. This produces a single graph (matrix), which makes the procedure less computationally demanding because the number of nodes is drastically decreased and just one minimum-cut algorithm is required. When more simulations are added to the process, the capacity of the arcs increases considerably.

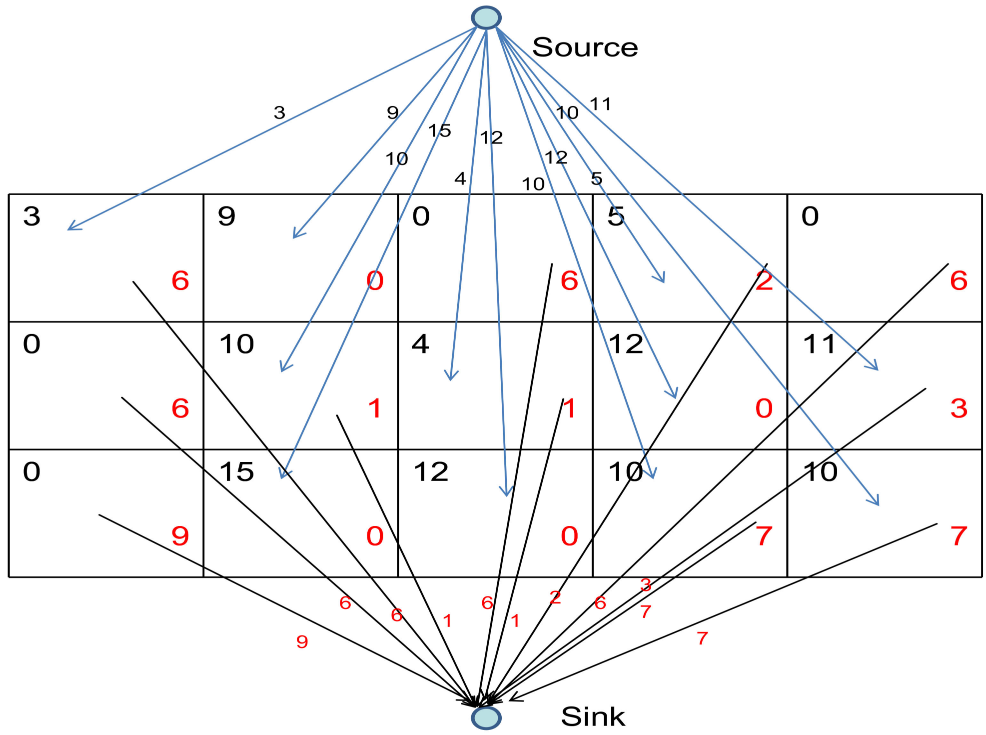

Figure 4. Three simulated models are combined to create new graphs.

3.3. Pushback Design Using Arc Capacity Parameterization in a Minimum-Cut Graph

The simulated orebody models construct a single pit, which is the ideal ultimate pit for the deposit, using the previously proposed minimum-cut network flow technique. Parameterization of the minimum-cut algorithm can be used to generate pushbacks. The Lerchs and Grossmann parameterization algorithm is well documented in the literature—for example, Seymour [50]—and is used in commercial implementations [23]. A succession of "nested" pits can be produced by scaling the economic values of all blocks with a multiplier parameter, λ. The same idea that may be used to build nested pits with a minimum-cut method can be used to parameterize the Lerchs and Grossmann algorithm.

Depending on which term is multiplied throughout the block economic value calculations, the value of λ must increase or decrease monotonically. If λ is used to immediately multiply the economic block value, rising λ values will produce pits ranging in size from small to large until the final pit limit is achieved. If λ is multiplied by the mining or processing cost, increasing values will result in larger to smaller pits. It is also feasible to utilize more than one parameter for parameterization. By changing the λ value to $\lambda_1, \lambda_2, \lambda_3, \ldots\ldots, \lambda_n$, where $\lambda_1 < \lambda_2 < \lambda_3 < \ldots\ldots < \lambda_n$, it is possible to obtain pushback $P_1, P_2, P_3, \ldots\ldots, P_n$ with pit sizes $P_1 < P_2 < P_3 < \ldots\ldots < P_n$. In this paper, λ was chosen as a monotonic non-decreasing value, which is multiplied by the capacity of the arcs from the source to the ore block nodes. Because the goal is to scale the economic worth of the ore blocks, the arcs from waste block nodes to sink are left alone. Figure 5 depicts the revised graph with the parametric version.

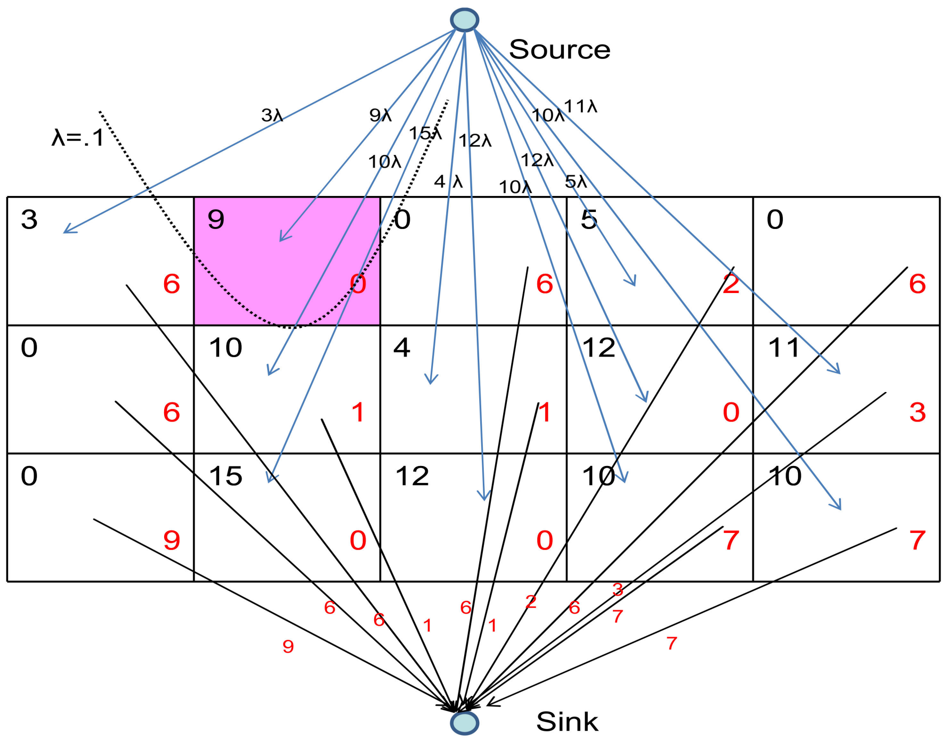

Figure 5. Parametric minimum-cut graph after multiplying λ with arcs from source to ore blocks. The first pushback is generated when the value of λ is 1.

It is obvious that the network algorithm's parameter λ, a multiplier of the arcs' capacity, is essential for creating nested pits or pushbacks. For the generation of nested pits, choosing a series of λ values is tough work. The random selection of λ may lead to pushbacks with huge gaps, which is unfavorable for mining. Additionally, a crucial issue is the number of pushbacks that must be created. For the sake of parameterization in this study, just one parameter is considered. When the λ value is set to 1, the ultimate pit is achieved. A sequence of nested pits is created by reducing the value of λ.

3.4. Mathematical Formulation of a Stochastic Mine Pit Optimization under Uncertainty

A two-stage stochastic mixed-integer programming model can be used to formulate the open-pit optimization issue. For each period of the horizon, a set of blocks to be mined is determined in the first stage, taking into consideration the minimum and maximum mining limits. Each block throughout each set is scheduled exactly once after all of its predecessors. The category of the block, such as ore or waste, and the amount of metal within the block are unknown at this point. For each scenario, uncertainty is addressed in the second stage. For instance, the total amount of ore needed for processing could, at times, surpass the processing plant's capacity, while at other times, it might not satisfy the minimum requirements. This may apply to the metal recovery from the blocks of processed ore. The suggested model's goal is to minimize the deviation from the production target while maximizing the profit for all simulations, S, by allocating N blocks over T production periods. The value of the objective function will be lower if the schedule deviates from the specified production target. The proposed model objective is presented in Equation (1). The objective must satisfy several constraint functions to meet the suggested model's goal. Constraint functions such as reserve constraints, slope constraints, mining constraints, processing constraints, and metal processing constraints are presented in Equations (2)–(6).

$$max\ \frac{1}{s}\{\sum_{s=1}^{S}\sum_{t=1}^{T}\sum_{i=1}^{N}c_{its}x_{it}-\sum_{s=1}^{S}\sum_{t=1}^{T}\left(v_t^{o-}d_{ts}^{o-}+v_t^{o+}d_{ts}^{o+}+v_t^{m-}d_{ts}^{m-}+v_t^{m+}d_{ts}^{m+}\right)\} \quad (1)$$

Subject to:

$$\sum_{t=1}^{T}x_{it}\le 1 \quad i=1,\ldots,N \quad (2)$$

$$x_{it}-\sum_{\tau=1}^{t}x_{p\tau}\le 0\ \ p\in P_i,\ \ t=1,\ldots,T \quad (3)$$

$$\begin{aligned}\sum_{i=1}^{N}w_ix_{it}\le \overline{W_{upper}}\ \ t=1,\ldots,T\\ \sum_{i=1}^{N}w_ix_{it}\ge \underline{W_{lower}}\ \ t=1,\ldots,T\end{aligned} \quad (4)$$

$$\begin{aligned}\sum_{i=1}^{N}o_{is}w_ix_{it}-d_{ts}^{o+}\le \overline{O_{upper}}\ \ s=1,\ldots,S\ ,t=1,\ldots,T\\ \sum_{i=1}^{N}o_{is}w_ix_{it}+d_{ts}^{o-}\ge \underline{O_{lower}}\ \ s=1,\ldots,S,\ t=1,\ldots,T\end{aligned} \quad (5)$$

$$\begin{aligned}\sum_{i=1}^{N}o_{is}m_{is}x_{it}-d_{ts}^{m+}\le \overline{M_{upper}}\ \ s=1,\ldots,S,\ t=1,\ldots,T\\ \sum_{i=1}^{N}o_{is}m_{is}x_{it}+d_{ts}^{m-}\ge \underline{M_{lower}}\ \ s=1,\ldots,S,\ \ t=1,\ldots,T\end{aligned} \quad (6)$$

where

c_{its} = Economic value of block i from simulation s for time t;

x_i = Mining block of an open-pit mine, where $x_i \in X$ and X is the set of all blocks in a deposit;

d = Discounted rate;

N = The total number of blocks considered for scheduling;

i = Block index, $i = 1, \ldots, N$;

T = The number of periods over which blocks are being scheduled;

t = period index, $t = 1, \ldots, T$;

P_i = The set of predecessors of block i; i.e., blocks that should be removed before i can be mined. Note that if block p is a predecessor of block i, then i is called a successor of p;

s_i = The set of successors of block i;

w_i = The weight of block i;

$o_{is} = \begin{cases} 1 \text{ if block } i \text{ is ore block in simulation } s \\ 0 \text{ otherwise} \end{cases}$

m_i = The amount of metal in block i;

S = The number of scenarios used to model geology uncertainties;

s = Scenario index, $s = 1, \ldots, S$;

$\overline{W_{upper}}$ = The maximum amount of material at period t;

$\underline{W_{lower}}$ = The minimum amount of material at period t;

$\underline{O_{lower}}$ = Minimum ore required to feed the processing plant during period t;

$\overline{O_{upper}}$ = Maximum ore processed in the plant during period t;

$v_t^{o-} = \frac{v^{o-}}{(1+d_2)^t}$ = Unit shortage cost that can be associated with failure to meet $\underline{O_t}$ during period t (v^{o-} is the undiscounted unit shortage cost, and d_2 represents the risk discount rate);

$v_t^{o+} = \frac{v^{o+}}{(1+d_2)^t}$ = Unit surplus cost incurred if the total weight of the ore blocks mined during period t exceeds $\overline{O_t}$;

$\underline{M_{lower}}$ = Minimum amount of metal that should be produced during period t;

$\overline{M_{upper}}$ = Maximum amount of metal that can be sold during period t;

$v_t^{m-} = \frac{v^{m-}}{(1+d_2)^t}$ = Unit shortage cost associated with failure to meet $\underline{M_t}$ during period t;

$v_t^{m+} = \frac{v^{m+}}{(1+d_2)^t}$ = Unit surplus cost incurred if the metal production during period t exceeds $\overline{M_t}$;

d_{ts}^{o-} = Shortage of ore at a discounted rate during period t in simulation s;

d_{ts}^{o+} = Surplus of ore at a discounted rate during period t in simulation s;

d_{ts}^{m-} = Shortage of metal for selling at a discounted rate during period t in simulation s;

d_{ts}^{m+} = Surplus of metal for selling at a discounted rate during period t in simulation s.

3.5. Solving Stochastic Graph Closure Problem for Pit Optimization

It must be noted that the target function of Equation (1) with the constraints of Equation (2) only provides the ultimate pit. A parametric graph is suggested in this research to meet the limitations of Equations (3)–(6). The network flow algorithm will be used to solve a parametric open-pit graph problem with the maximum flow or minimum cut. The maximum cut algorithm's purpose is to cut the arcs with the smallest capacity.

The proposed stochastic graph algorithm's parametric formulation is as follows:

$$\Phi(\lambda) = max\ \frac{1}{s} \sum_{s}^{S} \sum_{i=1}^{N} d_{i,s} x_i \tag{7}$$

where

$d_{i,s} = \lambda * c_{i,s}$ if $c_{i,s} > 0$; $d_{i,s} = c_{i,s}$ otherwise

$$\begin{gathered} x_i - x_p \le 0 \;\; i = 1, \ldots, N,\ p \in P_i \\ x_i = 0 \text{ or } 1 \;\; i = 1, \ldots, N, t = 1 \end{gathered} \tag{8}$$

The algorithm begins with a small λ value and updates it at each iteration.

3.6. Repair Algorithm to Generate a Feasible Solution

It is understood that the different λ values can generate different-sized pits; but there is no guarantee that the generated pit will respect the set of constraints that have dropped (Equations (3)–(6)). Without respecting this set of constraints, it is not possible to generate a feasible schedule for the production period. To generate a feasible schedule, the following steps will be followed:

Step 1: The problem formulated in Equations (7) and (8) will be solved with a small λ value, and the value of λ will be updated at each iteration until all lower limit constraints (Equations (4)–(6)) are violated for all simulations, S. Assign the solution of this step as $\Phi(\lambda_k)$, where k is the number of iterations.

Step 2: The problem formulated in Equations (7) and (8) will be solved with λ = λ_k value, and the value of λ will be updated at each iteration until all upper limit constraints (Equations (4)–(6)) are violated for all simulations, S. Assign the solution of this step as $\Phi(\lambda_m)$, where m is the number of iterations and *m* > *k*. A set of blocks, *j*, will be identified, such that $j \in \Phi(\lambda_m)$ and $j \notin \Phi(\lambda_k)$.

Step 3: The stochastic model will be formulated using these sets of j blocks incorporating the (Equations (4)–(6)) constraints. The updated stochastic model formulation can be presented as:

$$max \sum_{j=1}^{N_1} \sum_{s=1}^{S} c_{j,s} x_j \tag{9}$$

Subject to:

$$x_j \in \{0, 1\}, j \in N_1 \tag{10}$$

$$x_j - x_p \le 0 \;\; j = 1, \ldots, N_1,\ p \in P_j \tag{11}$$

$$\begin{gathered} \sum_{j=1}^{N_1} w_j x_j \le \overline{W_{upper}} - W1 \\ \sum_{j=1}^{N_1} w_j x_j \ge \underline{W_{lower}} - W1 \end{gathered} \tag{12}$$

$$\begin{gathered} \sum_{j=1}^{N_1} o_{js} w_j x_j - d_s^{o+} \le \overline{O_{upper}} - O1_s\ , s = 1, \ldots, S \\ \sum_{j=1}^{N_1} o_{js} w_j x_j + d_s^{o-} \ge \underline{O_{lower}} - O1_s\ , s = 1, \ldots, S \end{gathered} \tag{13}$$

$$\begin{gathered} \sum_{j=1}^{N_1} o_{js} m_{js} x_j - d_s^{m+} \le \overline{M_{upper}} - M1_s\ , s = 1, \ldots, S \\ \sum_{j=1}^{N_1} o_{js} m_{js} x_j + d_s^{m-} \ge \underline{M_{lower}} - M1_s\ , s = 1, \ldots, S \end{gathered} \tag{14}$$

where

N_1 is the number of blocks belonging to solution $\Phi(\lambda_m)$ but does not belong to $\Phi(\lambda_k)$;

$N_1 \le N,$ the computational time of SIP will be very less;

$W1 =$ Total weight of material in solution $\Phi(\lambda_k)$;
$O1_s =$ Total amount of ore at simulation s in solution $\Phi(\lambda_k)$;
$M1_s =$ Total amount of metal at simulation s in solution $\Phi(\lambda_k)$.

The minimum-cut technique will be applied to solve the following formulation, and the minimum-cut solution will be combined with $\Phi(\lambda_k)$ to generate a feasible solution. The proposed pseudo-code is used to achieve the goal of the proposed model.

Pseudo-code steps:

Step 1: Formulation of stochastic mine production scheduling of mining blocks, $x_i, x_i \in X$.

Step 2: Generate a graph problem (Φ_t) from a set time, t.

Step 3: Solve the graph closure problem using a minimum-cut algorithm.

Step 4: Choose parameter λ so that $\Phi(\lambda_k)$ violates all lower-bound constraints and $\Phi(\lambda_m)$ violates all upper-bound constraints; determine set j such that $j \in \Phi(\lambda_m)$ and $j \notin \Phi(\lambda_k)$.

Step 5: Solve the stochastic problem for the set j by the minimum-cut algorithm. (Solution = $\Phi(\lambda_m)$).

Step 6: Add $\Phi(\lambda_k)$ with $\Phi_\lambda = (\Phi(\lambda_m \cup \lambda_k)$ this is the solution of Φ_t.

Step 7: Eliminate $x_i, x_i \in \Phi_t$ from set X; new set $X = \{x_j, x_j \notin \Phi_t\}; t = t + 1$.

Step 8: If $t \leq T$, go to step 2 and repeat the process to find a new small pit.

Step 9: If $t \geq T$ stop the process.

4. Results

The process for the uncertainty-based ultimate pit limit and pushback design is demonstrated in this section using a case study. The suggested method for pushback design and final pit limit has been applied to a copper deposit. The deposit is situated in a greenstone belt that dates back to the Archean period. The region is primarily composed of mafic lavas, with minor amounts of moderate to felsic volcaniclastics. The geological dataset includes 185 drill holes with 10 m downhole composites in a 50 m $\times$ 50 m pseudo-regular grid, spanning a roughly rectangular region of 1600 $\times$ 900 m^2. A geostatistical analysis is used to define and model one mineralized domain using the geological information available. The simulation orebody model of the case study mine was generated using the direct block simulation technique [28]. The variogram analysis was used to determine the data's spatial correlation. The deposit's directional experimental variograms were determined in four different directions, with a spread of 22.5°, namely 0°, 45°, 90°, and 135°. The lag spacing for calculating the variograms along the strike for sulphides and oxides was 24 m and 32 m, respectively. Variograms along the downhole direction (−90° dip) were also calculated. All variograms were fitted with spherical variogram models. Except for the sulfide east along the downhole (dip −90°), all variograms were fitted by a single structure using a nugget model for copper. The deposit's block model was estimated using the ordinary block kriging estimation technique. Due to the size of the selective mining unit (SMU), the projected block size is 20 m $\times$ 20 m $\times$ 10 m. For this investigation, simulated models were created. Within the moralized zone, there are 9953 blocks with dimensions of 20 m $\times$ 20 m $\times$ 10 m. Equation (15) is used to calculate the block economic values of each block, which are then used to determine the ultimate pit limit and pushback design.

$$BEV = \begin{cases} Net\ revenue - MP - PP, & \text{if } Net\ revenue_i > PP \\ -MP_i, & \text{otherwise} \end{cases} \quad (15)$$

where

$$Net\ revenue = T \times G \times REC \times (\text{Price} - \text{Selling Price});$$

$MP=$ mining price, $PP =$ processing price, $T_i =$ tonnage, $G_i =$ grade, $REC =$ recovery

The economic parameters from Table 1 were utilized to determine the block's economic value.

Table 1. Economic parameters.

Parameters/Unit	Values
Copper price (US$/lb)	2.0
Selling price (US$/lb)	0.3
Mining price ($/tonne)	1.0
Processing price ($/tonne)	9.0
Recovery (%)	0.9

4.1. Ultimate Pit Generation

The directed graphs are created utilizing the block economic value of simulated orebody models to construct the ultimate pit. In this article, the block economic value is estimated as an undiscounted value. The simulation takes place in the mineralized zone; some waste blocks are added to the non-mineralized zone to establish a smooth topography and create a standard 3D orebody model. A directed graph is constructed, as mentioned in Section 3.2, with ore blocks linked to the source node and waste blocks linked to the sink node. To preserve slope restrictions, an infinite capacity arc is made for underlying blocks to overlying blocks. Since the study mine has a 45° slope angle, infinite capacity arcs are carried from an underlying block to nine overlying neighbor blocks. The infinite arc capacities are maintained by selecting a high positive number. The push re-label maximum flow algorithm is used to create the ultimate pit after the graph has been created. Two sections of the final pit created using the suggested method are shown in Figure 6.

Pushback Design and Mine Production Scheduling

The parameterization of the minimum-cut algorithm was carried out to produce a sequence of nested pits. The economic worth of the ore blocks was scaled using the parameter λ, as explained in Section 3.3, to produce eight pushbacks. To reduce the distance between pushbacks, the number of pushbacks in this study was determined through trial and error. Eight pushbacks were found to decrease the distance between them. The pushback sequences for the mine under study are shown in two sections in Figure 6.

It is necessary to evaluate the per-year mine production schedule to determine the discounted cash flows and total net present value for the case study. Production scheduling was carried out using the technical parameters listed in Table 1. Twenty simulated orebody models were used to create a model of the deposit, and from this model, the total ore quantity and ultimate pit limit were estimated to set the yearly production plan. The mine's life is estimated to be 8 years. As a result, the production goal was set by planning the model for 8 years. The considered cut-off grade for copper is 0.3%. To fulfill the production goals, bench-wise scheduling was used (i.e., extract the first bench of the first pushback, then the second bench of the first pushback, and so on, until the first pushback is extracted, and then repeat for the following pushback). Figure 7 displays the sections for an 8-year scheduling scenario. Figures 8–12 show the calculated (minimum, maximum, and average) mining capacity, ore production capacity, cumulative metal quantities, and comparison of the proposed model cumulative undiscounted cash flow with the deterministic model.

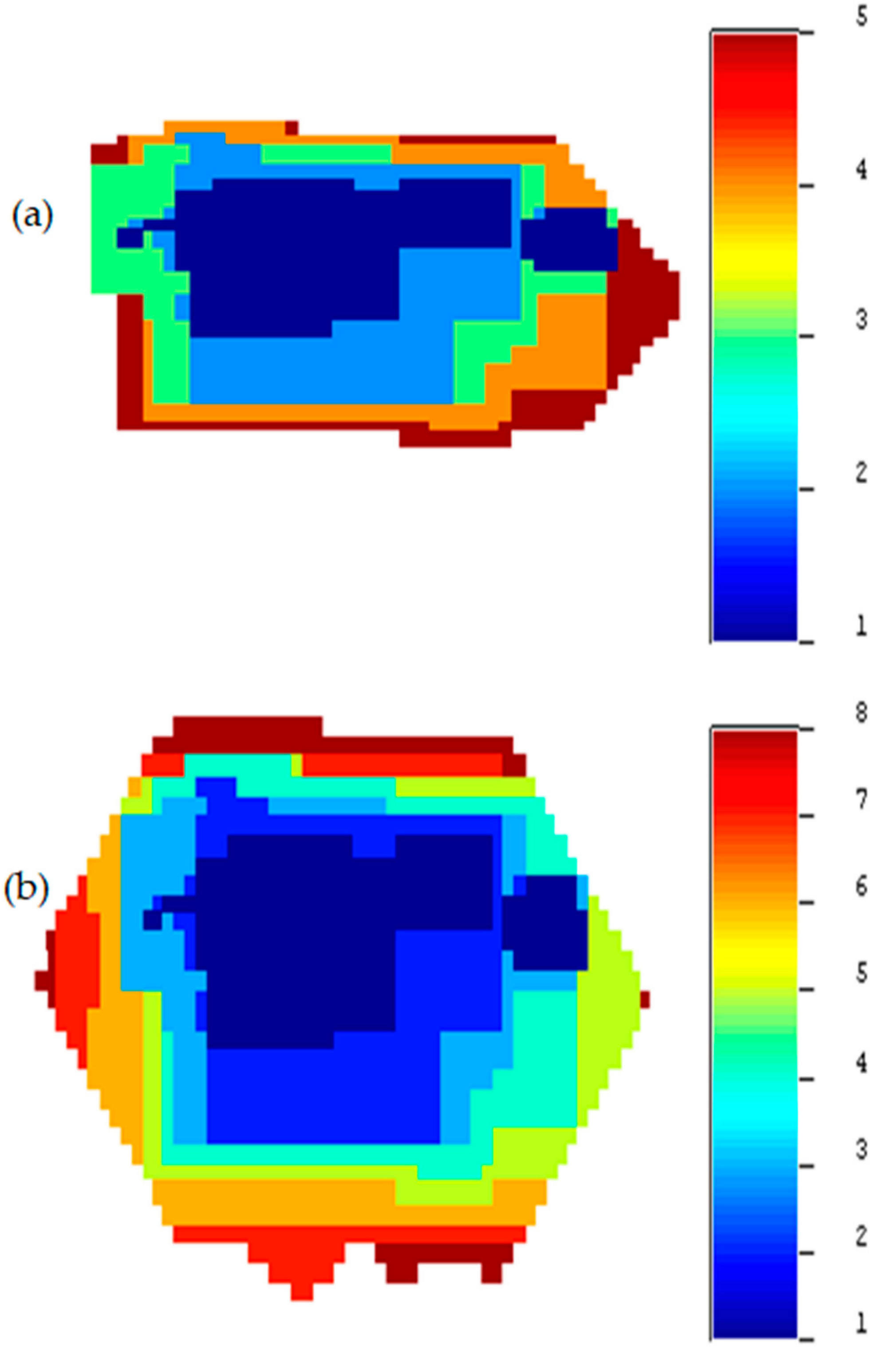

Figure 6. (**a**) Section view of the ultimate pit for 5 periods. (**b**) Top view of ultimate pit for 8 periods.

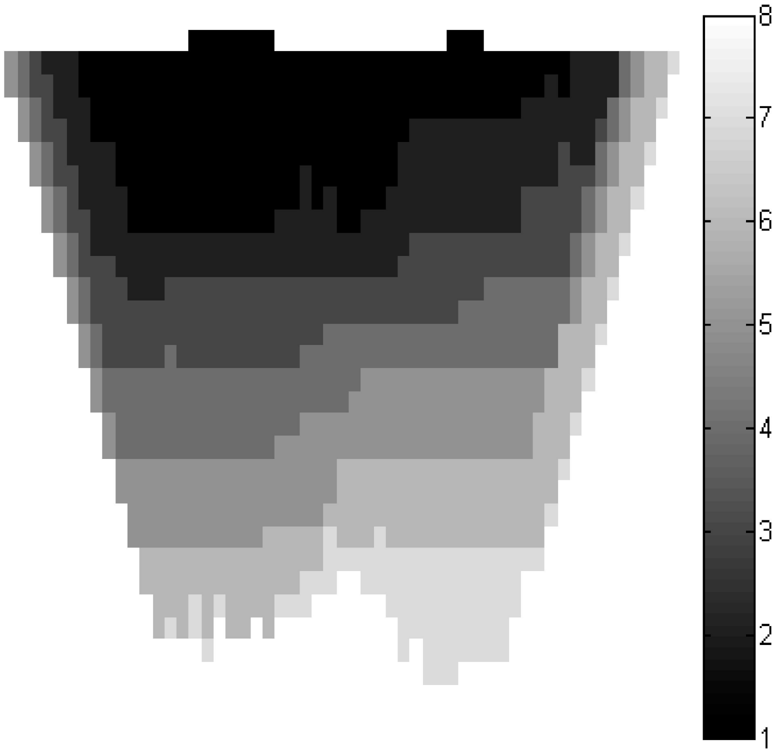

Figure 7. Production schedule of the deposit mine.

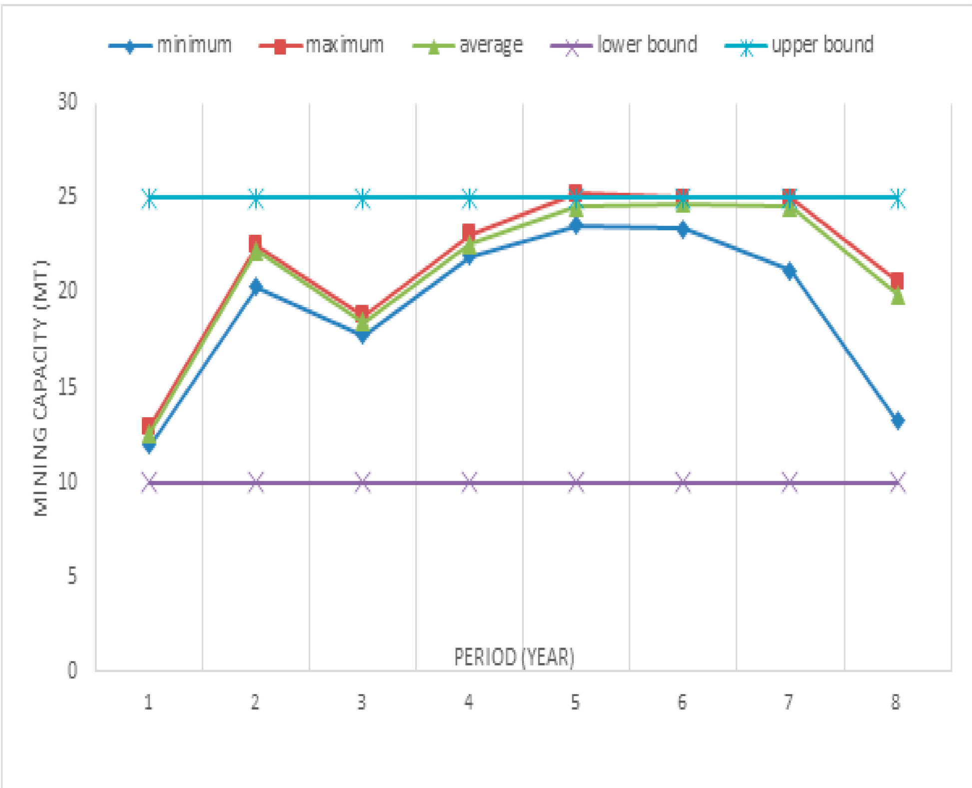

Figure 8. Mining capacity for production schedule.

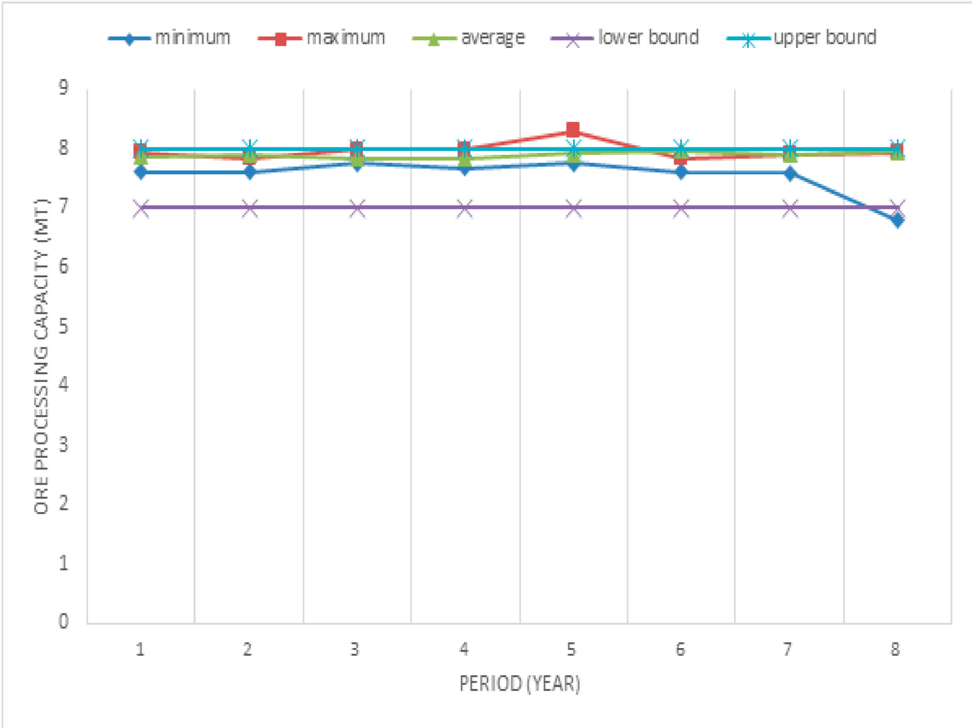

Figure 9. Ore production capacity of the production schedule.

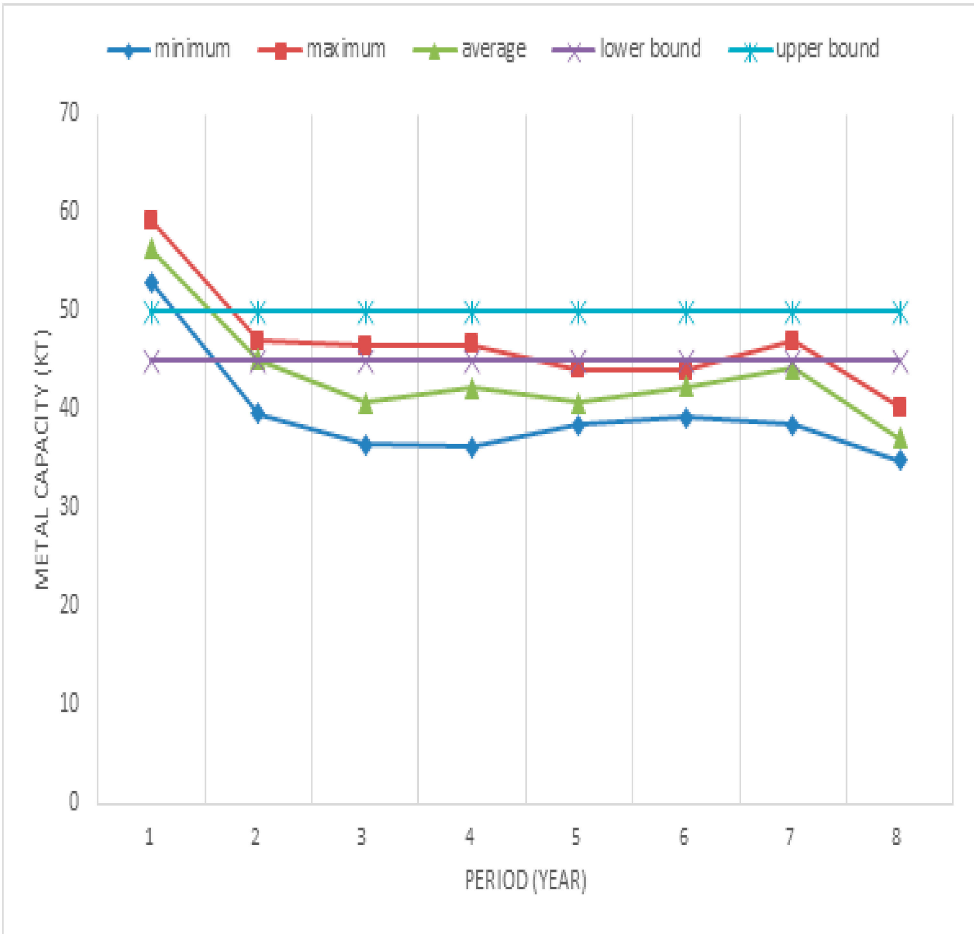

Figure 10. Metal production capacity of case study deposit.

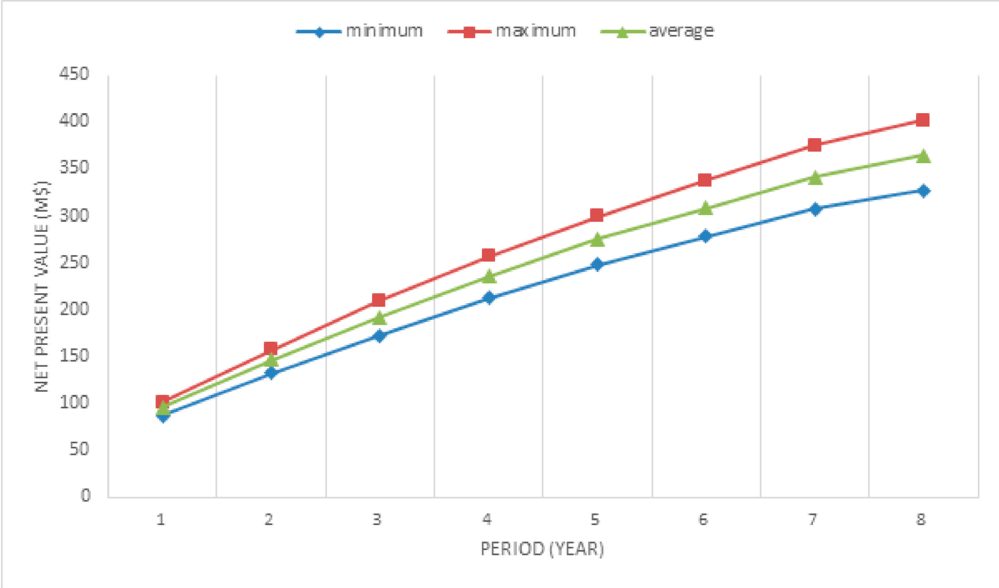

Figure 11. Undiscounted cumulative cash flow of case study deposit.

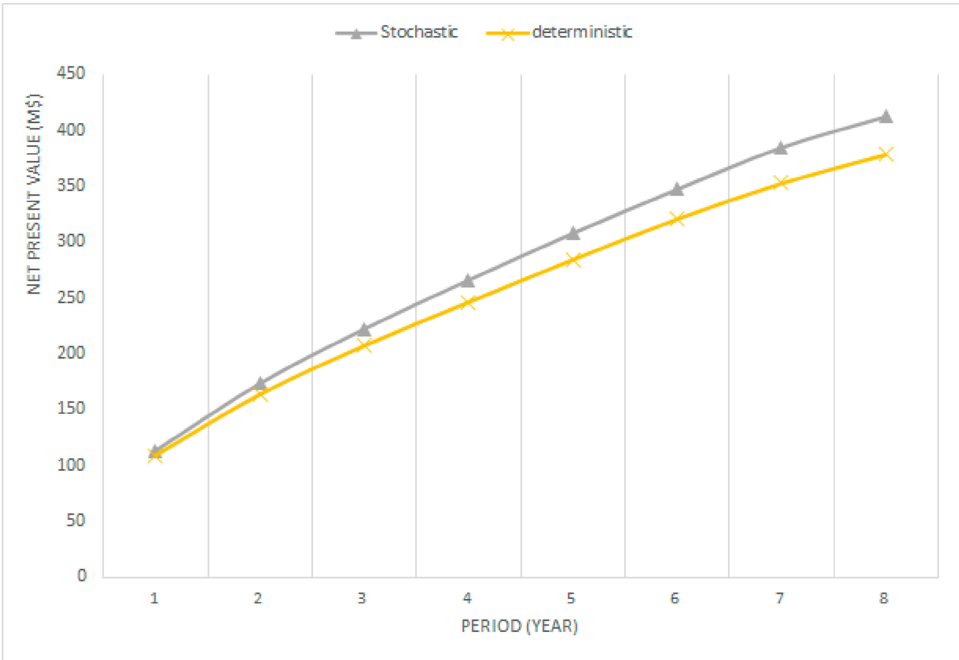

Figure 12. Comparison of undiscounted cumulative cash flow with a deterministic model.

5. Conclusions and Future Scope

To jointly integrate various orebody models, the minimum-cut graph approach was employed in this research together with the uncertainty-based ultimate pit limit and pushback design. The recommended method shows that it is quite simple to develop a minimum-cut network flow model that incorporates uncertainty and can handle uncertainty in the economic value of the mining blocks being scheduled. The example given here is based on block values that are uncertain because of the metal content, but the same method can be easily applied to account for demand uncertainties (commodity cost, exchange rate), as well as uncertainties in mining price, processing price, metal recovery, or any other input used to determine the block economic value. The key benefit of the suggested technique is that it is computationally very quick, making it possible to consistently incorporate many uncertainties in the optimization process. On a copper deposit with low variability, the suggested approach was successfully tested. The most significant conclusion was that, similar to other stochastic mine planning methods, the difference in net present value generated was 10% higher than with the equivalent deterministic (traditional) approach. We will run more field testing to examine the network flow approach's potential in uncertain situations.

Author Contributions: Conceptualization, D.J.; P.C.; A.Y.; methodology, D.J.; P.C.; A.S.; software, D.J.; P.C.; M.A.A.; validation, D.J.; P.C.; Y.M.; analysis, D.J.; P.C.; A.S.; investigation, Y.M.; resources, A.S.; M.A.A.; writing—original draft preparation, D.J.; P.C.; writing—review and editing, D.J.; P.C.; A.S.; A.Y.; supervision, A.S.; Y.M.; project administration, A.Y.; Y.M. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: 22UQU4400257DSR12.

Data Availability Statement: After signing a non-disclosure agreement, the data utilized in the study were obtained from a mining company. As a result, the data from the resource model cannot be shared. The solutions, however, can be shared. Please contact the primary author for clarification.

Acknowledgments: The authors would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: 22UQU4400257DSR12. and the support of NIT Rourkela and the mining industry for providing the environment and data to carry out the case study of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dagdelen, K. Open pit optimization—Strategies for improving economics of mining projects through mine planning. In Proceedings of the Seventeenth International Mining Congress and Exhibition of Turkey IMCET, Ankara, Turkey, 19–22 June 2001; pp. 145–148.
2. Hustrulid, W.; Kuchta, M.; Martin, R. *Open Pit Mine Planning and Design*, 3rd ed.; CRC Press: Boca Raton, FL, USA; Taylor and Francis: London, UK, 2013.
3. Kumar, A.; Chatterjee, S. Open-pit coal mine production sequencing incorporating grade blending and stockpiling options: An application from an Indian mine. *Eng. Optim.* **2017**, *49*, 762–776. [CrossRef]
4. Ramazan, S.; Dimitrakopoulos, R. Traditional and new MIP models for production scheduling with in-situ grade variability. *Int. J. Surf. Min.* **2004**, *18*, 85–98. [CrossRef]
5. Chatterjee, S.; Sethi, M.R.; Asad, M.W. Production phase and ultimate pit limit design under commodity price uncertainty. *Eur. J. Oper. Res.* **2016**, *248*, 658–667. [CrossRef]
6. Johnson, T.B. Optimum Open Pit Mine Production Scheduling. Ph.D. Thesis, Department of IEOR, University of California, Berkeley, CA, USA, 1968.
7. Gershon, M. Mine scheduling optimization with mixed integer programming. *Min. Eng.* **1983**, *35*, 314–329.
8. Caccetta, L.; Hill, S.P. An application of branch and cut to open pit mine scheduling. *J. Glob. Optim.* **2003**, *27*, 349–365. [CrossRef]
9. Ramazan, S. The new fundamental tree algorithm for production scheduling of open pit mines. *Eur. J. Oper. Res.* **2007**, *177*, 1153–1161. [CrossRef]
10. Bley, A.; Boland, N.; Fricke, C.; Froyland, G. A strengthened formulation and cutting planes for the open pit mine production scheduling problem. *Comput. Oper. Res.* **2010**, *37*, 1641–1647. [CrossRef]
11. Bienstock, D.; Zuckerberg, M. Solving LP relaxations of large-scale precedence constrained problems. In *IPCO 2010: Integer Programming and Combinatorial Optimization*; Lecture Notes in Computer Science; Eisenbrand, F., Shepherd, F.B., Eds.; Springer: Berlin/Heidelberg, Germany, 2010; Volume 6080, pp. 1–14.
12. Topal, E.; Ramazan, S. Strategic mine planning model using network flow model and real case application. *Int. J. Min. Reclam. Environ.* **2012**, *26*, 29–37. [CrossRef]
13. Chicoisne, R.; Espinoza, D.G.; Goycoolea, M.; Moreno, E.; Rubio, E. A New Algorithm for the Open-Pit Mine Production Scheduling Problem. *Oper. Res.* **2012**, *60*, 517–528. [CrossRef]
14. Cullenbine, C.; Wood, R.K.; Newman, A.M. A sliding time window heuristic for open pit mine block sequencing. *Optim. Lett.* **2011**, *5*, 365–377. [CrossRef]
15. Dimitrakopoulos, R.; Ramazan, S. Stochastic integer programming for optimizing long term production schedules of open pit mines: Methods application and value of stochastic solutions. *Min. Technol.* **2008**, *117*, 155–160. [CrossRef]
16. Lambert, W.B.; Newman, A.M. Tailored Lagrangian Relaxation for the open pit block sequencing problem. *Ann. Oper. Res.* **2014**, *222*, 419–438. [CrossRef]

17. Lamghari, A.; Dimitrakopoulos, R. A diversified Tabu search approach for the open-pit mine production scheduling problem with metal uncertainty. *Eur. J. Oper. Res.* **2012**, *222*, 642–652. [CrossRef]
18. Li, Y.; Topal, E.; Williams, D.J. Waste rock dumping optimisation using mixed integer programming (MIP). *Int. J. Min. Reclam. Environ.* **2013**, *27*, 425–436. [CrossRef]
19. Dowd, P.A. Risk assessment in reserve estimation and open pit planning. *Trans. Inst. Min. Metall. Sect. A Min. Ind.* **1994**, *103*, A148–A154.
20. Tachefine, B.; Soumis, F. Maximal closure on a graph with resource constraints. *Comput. Oper. Res.* **1997**, *24*, 981–990. [CrossRef]
21. Ravenscroft, P.J. Risk analysis for mine scheduling by conditional simulation. *Trans. Inst. Min. Metall. Sect. A Min. Ind.* **1992**, *101*, A104–A108.
22. Whittle, J. Beyond optimization in open pit design. In Proceedings of the Canadian Conference on Computer Applications in the Mineral Industries, Rotterdam, The Netherlands, 7–9 March 1988; pp. 331–337.
23. Whittle, J.A. Decade of open-pit mine planning and optimization—The craft of turning algorithms into packages. In Proceedings of the APCOM '99 28th International Symposium, Golden, CO, USA, 20–22 October 1999; pp. 15–24.
24. Dimitrakopoulos, R.; Farrelly, C.T.; Godoy, M. Moving forward from traditional optimization: Grade uncertainty and risk effects in open pit design. *Trans. Inst. Min. Metall. Sect. A Min. Technol.* **2002**, *111*, A82–A88. [CrossRef]
25. Godoy, M.; Dimitrakopoulos, R. Managing risk and waste mining in long-term production scheduling of open-pit mines. *SME Trans.* **2004**, *316*, 43–50.
26. Goodfellow, R.; Dimitrakopoulos, R. Algorithmic integration of geological uncertainty in pushback designs for complex multi-process open pit mines. *Min. Technol.* **2013**, *122*, 67–77. [CrossRef]
27. Goodfellow, R.; Dimitrakopoulos, R. Simultaneous Stochastic Optimization of Mining Complexes and Mineral Value Chains. *Math. Geosci.* **2017**, *49*, 341–360. [CrossRef] [PubMed]
28. Boucher, A.; Dimitrakopoulos, R. Block simulation of multiple correlated variables. *Math. Geosci.* **2008**, *41*, 215–237. [CrossRef]
29. Goovaerts, P. Geostatistics for natural resources evaluation. In *Applied Geostatistics Series*; Oxford University Press: New York, NY, USA, 1997.
30. Mustapha, H.; Dimitrakopoulos, R. High-order stochastic simulation of complex spatially distributed natural phenomena. *Math. Geosci.* **2010**, *42*, 457–485. [CrossRef]
31. Minniakhmetov, I.; Dimitrakopoulos, R.; Godoy, M. High-order spatial simulation using Legendre-like orthogonal splines. *Math. Geosci.* **2018**, *50*, 753–780. [CrossRef] [PubMed]
32. Ramazan, S.; Dimitrakopoulos, R. Stochastic optimization of long-term production scheduling for open pit mines with a new integer programming formulation. *Orebody Model. Strateg. Mine Plan.* **2007**, *14*, 359–365.
33. Boland, N.; Dumitrescu, I.; Froyland, G. A Multistage Stochastic Programming Approach to Open Pit Mine Production Scheduling with Uncertain Geology. *Optimization* **2008**, 1–33. Available online: https://optimization-online.org/2008/10/2123/ (accessed on 19 April 2022).
34. Leite, A.; Dimitrakopoulos, R. A stochastic optimization model for open pit mine planning: Application and risk analysis at a copper deposit. *IMM Trans. Min. Technol.* **2007**, *116*, 109–118. [CrossRef]
35. Albor Consuegra, F.R.; Dimitrakopoulos, R.G. Stochastic mine design optimisation based on simulated annealing: Pit limits, production schedules, multiple orebody scenarios and sensitivity analysis. *Min. Technol.* **2009**, *118*, 79–90. [CrossRef]
36. Lerchs., H.; Grossmann, I.F. Optimum Design of Open Pit Mines. *Trans. CIM* **1965**, *58*, 47–54.
37. Ramazan, S.; Dimitrakopoulos, R. Production Scheduling with Uncertain Supply: A New Solution to the Open Pit Mining Problem. *Optim. Eng.* **2013**, *14*, 361–380. [CrossRef]
38. Montiel, L.; Dimitrakopoulos, R. A heuristic approach for the stochastic optimization of mine production schedules. *J. Heuristics* **2017**, *23*, 397–415. [CrossRef]
39. Rimélé, A.; Dimitrakopoulos, R.; Gamache, M. A stochastic optimization method with in-pit waste and tailings disposal for open pit life-of-mine production planning. *Resour. Policy* **2018**, *57*, 112–121. [CrossRef]
40. Godoy, M. The Effective Management of Geological Risk. Ph.D. Thesis, University of Queensland, St Lucia, QLD, Australia, 2003.
41. Remy, N.; Boucher, A.; WU, P. *Applied Geostatistics with Sgems—A User's Guide*; Cambridge University Press: Cambridge, UK, 2009.
42. Dimitrakopoulos, R.; Ramazan, S. Uncertainty-based production scheduling in open pit mining. *SME Trans.* **2004**, *316*, 106–116.
43. Dimitrakopoulos, R.G.; Grieco, N. Stope design and geological uncertainty: Quantification of risk in conventional designs and a probabilistic alternative. *J. Min. Sci.* **2009**, *45*, 152–163. [CrossRef]
44. Ramazan, S.; Dimitrakopoulos, R. Stochastic Optimisation of Long-Term Production Scheduling for Open Pit Mines with a New Integer Programming Formulation. In *Advances in Applied Strategic Mine Planning*; Springer: Berlin/Heidelberg, Germany, 2018. [CrossRef]
45. Albor Consuegra, F.R.; Dimitrakopoulos, R. Algorithmic Approach to Pushback Design Based on Stochastic Programming: Method, Application, and Comparisons. *IMM Trans. Sect. A Min. Technol.* **2010**, *119*, 88–101. [CrossRef]
46. Chatterjee, S.; Dimitrakopoulos, R. Production scheduling under uncertainty of an open-pit mine using Lagrangian relaxation and branch-and-cut algorithm. *Int. J. Min. Reclam. Environ.* **2020**, *34*, 343–361. [CrossRef]
47. Moreno, E.; Emery, X.; Goycoolea, M.; Morales, N.; Gonzalo, N. A Two-Stage Stochastic Model for Open Pit Mine Planning under Geological Uncertainty. 2017. Available online: http://mgoycool.uai.cl/papers/17moreno_apcom.pdf (accessed on 19 April 2022).

48. Koushavand, B.; Askari-Nasab, H.; Deutsch, C.V. A linear programming model for long-term mine planning in the presence of grade uncertainty and a stockpile. *Int. J. Min. Sci. Technol.* **2014**, *24*, 451–459. [CrossRef]
49. Hochbaum, D.S. A new-old algorithm for minimum cut in closure graphs. *Netw. Spec. 30th-Anniv. Pap.* **2001**, *34*, 171–193.
50. Seymour, F. Pit limit parameterisation from modified 3D Lerchs-Grossmann Algorithm. *SME Trans.* **1995**, *298*, 1860–1864.

ANN-ABC META-HEURISTIC HYPER PARAMETER TUNING FOR MAMMOGRAM CLASSIFICATION

Article *in* Journal of Theoretical and Applied Information Technology · December 2022

2 authors:

Ajay Kumar Mamindla
Osmania University

Ramadevi Yellasiri
Chaitanya Bharathi Institute of Technology

Journal of Theoretical and Applied Information Technology
31st December 2022. Vol.100. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ANN-ABC META-HEURISTIC HYPER PARAMETER TUNING FOR MAMMOGRAM CLASSIFICATION

AJAY KUMAR MAMINDLA[1], DR. Y. RAMADEVI[2]

[1]Research Scholar, Department of CSE, Osmania University, India
[2]Professor, Department of CSE, CBIT, India

E-Mail: [1]ajaymamindla@gmail.com, [2]yramadevi_cse@cbit.ac.in

ABSTRACT

In recent past, artificial neural networks (ANN) have reaped improvements in the domain of medical image processing by addressing many unmanageable problems. The initialized hyperparameters control ANN performance and selecting sensible hyperparameters by hand is time-consuming and tiresome. This study suggests a metaheuristic optimization of the fine-tuning hyperparameters approach to remedy this flaw. The method is then evaluated on mammography images to assess whether the mammogram contains cancer. In the proposed ANN model, a modified Artificial bee colony (ABC) optimization method is used to fine tune the hyperparameters, and it categorizes the tumors in the breast as benign or malignant in two-class case and normal, benign, and malignant in three-class case with an accuracy of 97.52% and 96.58% respectively. Hyperparameters to the neural network framework were assigned instantly with the help of ABC method with wrapped ANN as objective function. Manual search, Grid Search, Random Grid search, Bayes search are all cutting edge ANN hyperparameters methods. In addition to the mentioned, nature-inspired optimization methods such as PSO and GA have adopted for fine tuning parameters. Additionally, the suggested model's performance in classifying breast pictures was compared to that of the published hyperparameter technique using sizable datasets on breast cancer that were made accessible to the public.

Keywords: *Artificial Neural Networks, Hyperparameters, Artificial bee colony, Mammogram images, Grid Search.*

1. INTRODUCTION

Out of two women newly breast cancer diagnosed one woman dies in India. Breast cancer ratio is 14% of all women cancers in India and in it is the topmost cancer in case of new cases registered in 2020 with 178361(26.3%) cases in women in India [25]. Overall, out of twenty-nine women one woman likely to detect with Breast cancer in her lifetime. Furthermore, it the second most common cancer in the world next to lung cancer, irrespective of gender [1]. Studies are proved that early breast cancer detection could cut the rate of death drastically, reduces the radiologist effort to treatment and disease morbidity [2]. For the Radiologist, early state breast cancer detection is a tiresome job as they must deal with huge number of digital mammogram images, which inclined to develop an automated and simplified early-stage cancer detection method. Medical diagnosis system could have been inculcated with the intelligent based systems like neural networks models [24]. Traditional decision-making methods are outperformed by the artificial intelligence-based ANN models. In this regard, many swarm based nature-inspired optimization algorithms (NIOA) like ABC, PSO (particle swarm optimization) and GA (Genetics Algorithm), etc. are adopted in finding the optimized solutions to real-time medical diagnosis problems [3]. This research proposed an automated hyperparameter fine tuning ANN model by adopting ABC as its tuning method instead of traditional methods like Grid Search.

1.1 Artificial Neural Networks

Biological neurons in human brain and their relationships inspire the development of basic ANN model. ANN metrics like performance accuracy and efficiency builds upon its structural parameters like number of neurons in input and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

output layers, count of hidden layers, activation function and weighted values. Generally, multilayer ANN have one input, one output and at least one hidden layer [4]. Figure one depicts the basic structure of ANN.

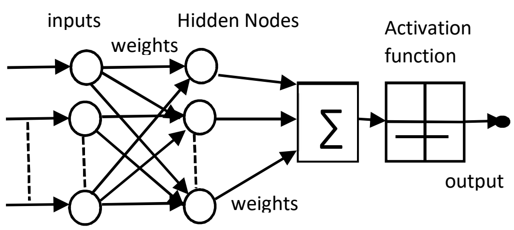

Figure 1: ANN Basic Structure [4].

1.1.1 ANN parameters vs hyperparameters

Decision making has huge involvement of Machine Learning (ML) models. In real time problems, the professionals working around with various algorithms depends upon the problem and the selected model. However, professional can increase the accuracy and performance of the ML model using hyperparameters. ML models are associated with both hyperparameters and parameters and both are intended for distinguished tasks. ML model estimates the parameters based on the input data history for result prediction. One cannot hardcode or set the parameters like independent variables coefficient, logistic regression, and linear regression to a ML model. Whereas professionals can specify some variables to the ML model manually those are called hyperparameters. Hyperparameters helps in evaluating the best optimal parameters to the model and these are decided by the one who builds model. Random forest algorithm max_depth, KNN classifiers k value, number of neurons in NN layers, etc. are some hyperparameter examples [26]. Table 1 shows the optimal range of ANN hyperparameters [5].

1.1.2 hyper parameter methods

Optimization of hyper parameter is achieved with various optimization methods. This section listed most popular four parameter optimization methods. 1. Manual search 2. Grid search [GS] 3. randomized search [RS] 4. Bayes Search [BS].

Table 1: ANN Model Hyperparameters and Their Ranges [5].

S.No	Hyperparameter	Description	Range
1	No. of Hidden Layers	between the input and output layers, the number of inner layers	1 to 3
2	Number of hidden nodes	in the hidden layer, the number of neurons	1 to 10
3	Number of training cycles	No. of the training iterations	10 to1000
4	Learning Rate	Weight variation updated during learning	0.0001 to 0.1
5	Learning algorithm	The learning process is carried out via an optimization algorithm in a neural network.	RMDprop, SDG, Adam,
6	Adam, SDG, RMDprop	activation function of Neurons	Linear, Tangent
7	Learning rate decay	The decay of learning rate across learning iterations' rate function	linear ,Exponential
8	Error function	the process through which a neural network's training function is reduced	mean square error, Log loss
9	Epoch limit	Maximum number of iterations for learning	Maximum number of learning iterations
10	Mini batch size	Group size submitted to model during training	10, 20, 30
11	Patience	a delay to the trigger in terms of how many epochs we'd prefer to go without improvement.	2, 5, 10

In Manual Search, the professional's expertise or intuition determines the required hyperparameters values for NN, but one who can set parameters has excellent grasp on learning data and NN structure. However, several trial-and-error experimentations leads to setting wise hyperparameters of the model. GS identifies the suitable and well performed hyperparameters by combining parameters and using many values for hyperparameters while calculation. GS needs minimal or less background knowledge, easy to apply and straightforward method. In GS, Lower and Upper limits of all hyperparameters are used to explore the potential and possible hyperparameter combinations and identify the best parameter set of models. As a result, GS creates a value space of hyperparameter in the predefined step. GS is recognized as a broad space as it executes all possible and potential combinations. Thus, GS is more time complex method and cost of computation is high. Moreover, GS gives different performances to the same set of parameters, when applied in ANN, Convolutional NN and Recurrent NN [5]. RS is same as GS method, whereas in RS comprehensive enumeration of parameters combinations are replaced by random selection of hyperparameters. RS with less cost, explores the more search space compared to GS. Major advantage of RS is independent evolutions are parallelized with ease and allocated resources effectively [6]. Furthermore, global optimization method BS is used erroneous black box functions. Bayesian optimization creates a mapping function's probabilistic model to map the validation set assessment objective to hyperparameter values. Figure 2a depicts that GS applying varying values of two hyperparameters. For the sum of hundred feasible combinations, every hyperparameter is compared and checked with 10 definite values. Blue and Red outlines denote the regions with robust outcomes and places with low results respectively. Figure 2b depicts that RS considered the hundred different options to do a random searching across the values of possible combinations for two hyperparameters. When compared to GS, RS examined a greater number of each hyperparameter values separately. Figure 2c illustrate that BS examine all hyperparameters alternate options by identifying next combination to be examined using past discoveries [7].

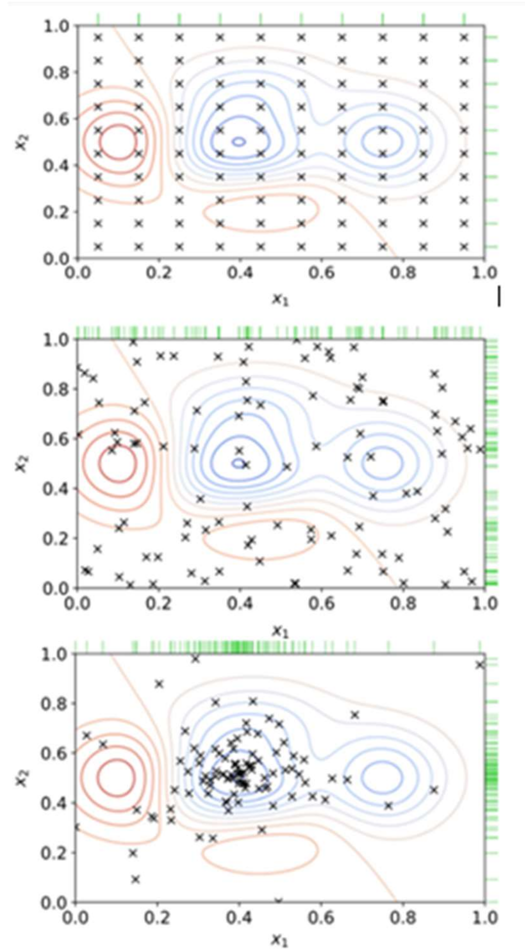

Figure 2: A. Grid Based Search B. Random Search C. Bayesian Search [7].

1.2 Nature Inspired Hyperparameter tuning

1.2.1 Genetics Algorithm

GA is evaluation theory-based metaheuristic algorithm. Every generation generates worse and superior individuals by inheriting traits from parents. In the long run, superior will persist and worse will gradually perish. After several generations, the best adaptable item will be the potential global optimum. Which means the entity with capable to adapt to surroundings and excellent survival capacity are likely to get over through the traits and live on. Figure 3 depicts the genetic algorithms template in four step methodology [11].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

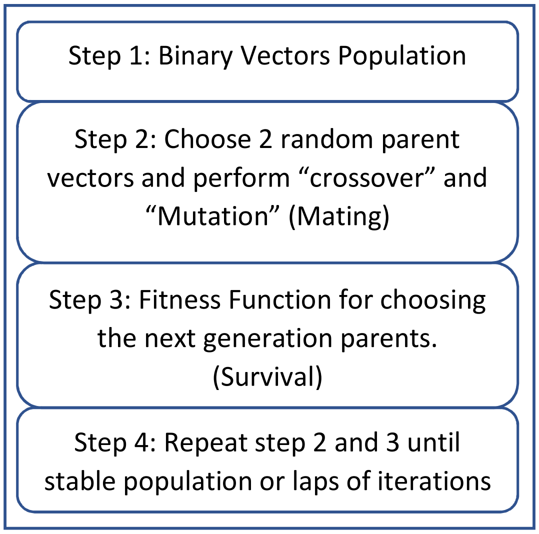

Figure 3: Genetics Algorithm Template [11].

1.2.2 Particle swarm optimization

PSO is initialized with a random solutions population, which is similar to GA. Whereas, PSO differs from GA algorithm in the system every possible solution is assigned with a randomly generated velocity and later these particles or potential solutions are flown in hyperspace. In PSO three best values called *best*, *pbest* and *gbest* are keep tracked. *best* solution is the coordinates in hyperspace achieved thus far are keep tracked by every particle. *pbest* is the fitness value of *best* solution. *gbest* is the location of best value obtained by any particle so far in the population [8]. The process of PSO is described as follows, in each iteration particle update its accelerating velocity towards its global version *gbest* and *pbest*. Acceleration is updated by a separately generated random number and a random term toward *gbest* and *pbest*.

This research main contributions could well be summarised as follows:

- *This research proposes integration of ANN with NIOA Hyperparameter Methods*
- *Proposed method is employed to Mammograms Classification using ABC based ANN.*
- *The mammogram classification results are compared and tested with other state-of-art methods in 2-class and 3-class ways.*

Artificial Bee Colony optimization

ABC is bio-inspired meta-heuristic optimization method, firstly materialized by Karaboga in 2005[13]. ABC mimic the foraging behavior of hive honeybees. Because of its low control parameter, simplest design model and good robustness takes over the other NIOA optimization algorithms among the young researchers. The process of basic ABC is described with foraging behavior, food sources, employee, onlooker, and scout bees [12]. ABC algorithm is divided into mainly four different phases. 1. Initialization Phase: scout bee specifies the population of food sources and control parameters. 2. Employee Bee: after finding a potential solution it searches for the nearby areas by comparing the fitness value of food source (Equation 1). 3. Onlooker Bee: employee and onlooker bees exchange the information about food sources in the dancing area of hive (Equation 2) 4. Scout bee: trial or limit exhausted food sources are abandoned by the scout bee using upper and lower bounds (Equation 3). Nectar amount and position of food source calculate the fitness value [3].

$$X_{new}^{j} = X^{j} + \emptyset(X^{j} - X_{P}^{j}) \tag{1}$$

Where 'j' is food source randomly selected, ∅ is a value between (-1, 1) generated randomly, and p partner food source selected randomly [12].

$$Prob_i = 0.9 * \frac{Fit_i}{\sum_{1}^{n} Fit} + 0.1 \tag{2}$$

Where 'Fit_i' specific solution's objective function value. X_k randomly generated new solution [12].

$$X_k = lb + (ub - lb) * r \tag{3}$$

2. RECENT RESEARCH

Roseline et all [2022], proposed a medical diagnostic system by employing the NIOA method called PSO to fine tune the hyperparameters of ANN and CNN. The WDBC (Wisconsin Diagnostic Breast Cancer) data set was tested with the proposed methodology and achieved the classification accuracy of 99.2% and 98.5% for ANN and CNN respectively [5]. Vishnu et al [2022], explores how can an ANN model predict the shear walls failure models using hyperparameter optimization and achieved

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

comparatively good results. This study practiced the fine tuning of ANN hyperparameters using GS, RS, BS, HyperBandCV, PSO and GA methods [6]. Zhiqiang Guo et al[2022], aims at Multi-layer Perceptron (MLP) NN hyperparameter tuning to classify the samples of breast cancer. This proposed ensemble learning algorithm is intended to improve the NN performance by combining the traditional parameter optimization methods with the NIOA algorithms like GA, PSO and ODMA. In this study, comparison of different algorithms is made on three breast cancer datasets to achieve 98.79% of accuracy [14]. Warut Pannakkong et all [2022], used response surface methodology (RSM) as hyperparameter methos instead of most used GS method. This study applied RSM in fine tuning of three ML algorithms: ANN, deep belief network (DBN) and SVM. Proposed approach RSM for ANN, DBN and SVM outperform the GS for ANN, DBN and SVM in prediction accuracy, number or runs and settings of hyperparameter with large margin [15]. Punitha et al [2021], proposed a hybrid method for hyperparameter tuning by employing ABC and Whale Optimization (WHO). ABC phases are inculcated by inheriting the attacking behaviour of whale. WDBC, MIAS etc. are used to test the proposed method and got the 99.2% of accuracy using hybrid method [16]. Punitha S et al [2021], proposed IAIS-ABC-CDS (integrated Artificial immune system and-ABC-Cancer Diagnosis) method for ANN parameter optimization and feature selection. This method improves the local search process by taking the simulated annealing (SA) method advantages. WDBC data set was taken for testing the proposed methodology and it is reported that 99.11% and 99.34% mean classification in ANN [17]. siti Fairuz mat radzi et al [2021], developed a Automated ML (AutoML) method as an pipeline optimization technique for identifying ML models with excellent performance and simple pipelines for breast cancer diagnosis. Moreover, the presented method exceeded the GS performance in optimization process. The proposed classifier results the good outcome compared to other reported models with only couple of preprocessors and with 0.83 precision values [18]. Gamz Erdogan Erten et al [2020], developed a GS based ANN hyperparameter method. This method performance is compared with KNN (k-nearest nighbour), Naïve Bayes (NB), Support vector Machine (SVM) and decision tree (DT). It produced good accuracy values as 73.95%, 76.74%, 77.21%, 81.86%, 91.63% with the GS based KNN, NB, DT, SVM and ANN methods respectively [19]. Fernando Itano et all [2018], developed an optimization search method for MLP method using GA. It examines the classification performance with hyperparameter correlation by allowing the less search space. It added hyperparameter for the initialization and regularization of weights simultaneously with learning of Hyperparameter and MLP topology [20]. P. Shanmugapriya et al [2017], presented a novel Swarm Intelligence based hybrid method called ABC-AC (ant colony) for feature selection for the image classification. This study takes the advantages of both ABC and AC optimization algorithm to achieve the better results. Proposed method is applied on different decease datasets [21]. Fadzil Ahmad et al [2012], proposed an automated BC diagnosis method using GA for ANN feature selection and optimization of hyperparameters. This method is developed in three variations like GA_ANN_RP (resilient Back-propogation(BP), GAANN_LM (Levengerg Marquardt) and GAANN_GD (gradient descent) for ANN parameter and weight optimization. Interestingly, GAANN_RP is tops among the three proposed methods with the 99.24 % of average and 98.29% of classification correctness [22].

3. PROPOSED ANN-ABC METHODOLOGY

3.1 Problem statement

According to several researchers, adjusting the ANN parameters can increase model accuracy while lowering time complexity. The relevant investigation that was mentioned in the preceding section concluded that the best strategy for parameter tweaking was still needed. The primary goal of this study is to use NIOA's ABC method to fine-tune the ANN model's hyperparameters. The population space, search area, duration, and spatial complexity of traditional optimization methods each have their own pros and downsides. This paper addresses the some of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

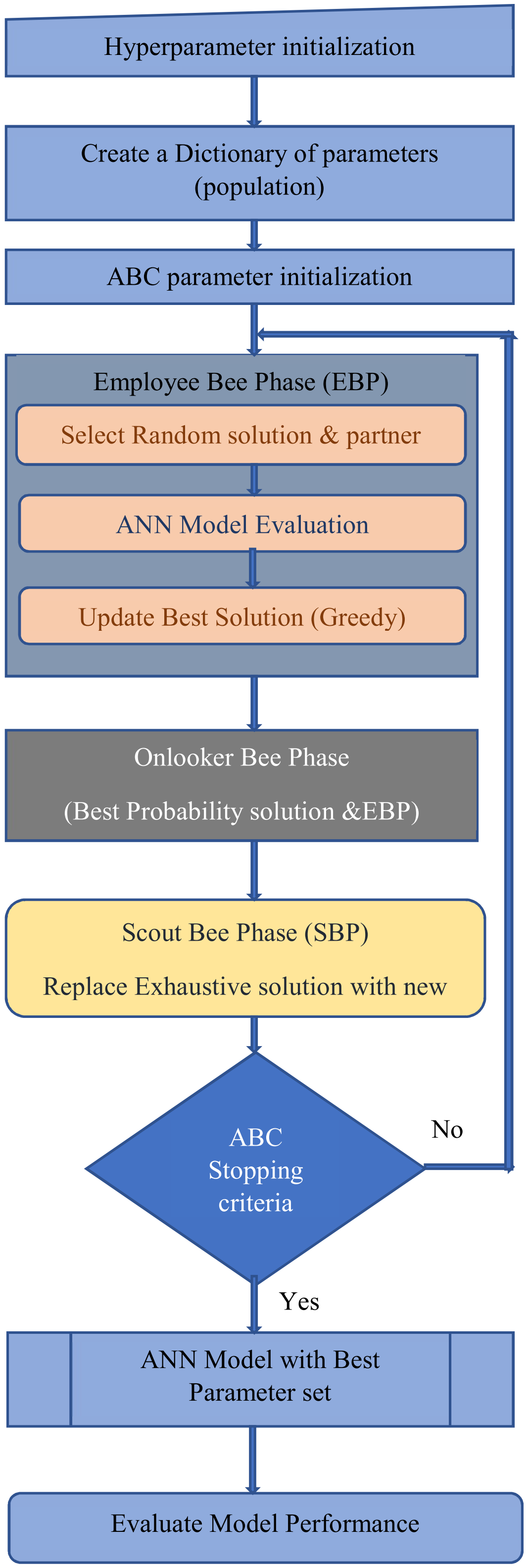

Figure 4: Proposed Methodology.

mentioned problems like time and search space complexity with reduced search space, exploration, and exploitation by adopting the NIOA ABC algorithm.

3.2 Overall design

Figure 4 illustrates the flow diagram of the proposed ANN-ABC model for hyperparameter tuning. The proposed method has primarily three steps: 1. Parameter initialization and population creation 2. Apply ABC parameterization to ANN model 3. Assess the accuracy and complexity of the best solution set obtained in the previous step. Firstly, ANN hyperparameters like the number of hidden layers, number of neurons in each layer, learning rate, activation functions, etc. are to be initialized with the most popular optimal set derived from the earlier research. After that dictionary of specified parameters is arranged as a dictionary, it is sent as a population space to the ABC phase. Second, in this step, a number of employee bees is assigned to the number of inputs, and they start exploring nearby locations randomly and comparing the fitness values. Once the greedy solution is discovered, it applies that solution set to the wrapped ANN model (passed as an objective function to ANN) and determines the model's accuracy. Later, by comparing the fitness values of solutions, the OBP determines the probability of the best solution being returned by the EBP and performs greedy selection. In SBP, if any solution is considered to have more than the limit of occurrences, then it is replaced with the newly randomly selected solution. The above process is iterated till it reaches the specified number of iterations or stopping condition. Furthermore, the ABC phase returns the best solution set of hyperparameters to the ANN model. Finally, tuned hyperparameters are again applied to determine the accuracy and complexity of the model.

4. RESULT AND DISCUSSIONS

4.1 Dataset: Experimental setup

Proposed classifier model performance is investigated by using an open source mini-MIAS (Mammogram Image Analysis Society) data set. The database involves a set of 322 mammogram digital images comprises 62 benign, 51 malignant and 209 normal mammograms [23]. Proposed

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

classifier is implemented in two ways, three-class case (normal, benign, and malignant) and two-class case (normal and abnormal). In two class-case, proposed classifier is applied on dataset to categorize mammogram into normal and abnormal. Whereas, in three-class set, classifier categorize the images into normal, benign, and malignant. Out of 322 images, 70% mammogram are considered as training set and 30% as testing set and used k-fold cross-validation (CV). The input data is normalized by using scikit learn minimum-maximum scaler and the categorical variables are passed to ANN input layer after one hot encoding.

The proposed model performance is evaluated by using the metrics like specificity(recall), F-Score, sensitivity, precision, and accuracy. These metrics are calculated from:

Specificity = TN / (FP + TN)

Sensitivity (Recall) = TP / (FN+TP)

Accuracy=(TP+TN)/(FP+FN+TP+TN)

Precision =TP/(TP+FP)

FScore=(2*Precision*Recall)/(Precision+Recall)

4.2 ANN-ABC Hyperparameter setup

ANN model is wrapped as a single function and passed the set of hypermeters as parameter to the wrapped function. Further, the dictionary of hyperparameter is prepared with different parameters, such as the hidden layers count, every hidden layer's neuron count, activation functions, learning rate, dropout size, epochs (as ABC termination), and batch size. In this practice, ADAM algorithm, binary_crossentropy used as optimizer and loss functions while compiling ANN model. The created Dictionary grid space is considered as the ABC population and size of bees is depending on the count of hyperparameters used. Same parameter set is applied to all hyperparameters methods like GS, RS, BS, PSO, GA and proposed ANNABC models in two ways like two-class case and three class case. The performance of all mentioned models is compared and illustrated as confusion matrix and its metrics. Table 2 show the set of hyperparameters used for the experiments.

Table 2: Hyperparameter Dictionary

Parameter	Count	Range
Hidden Layers	1,2,3	[20], [40,20],[40,30,15]
Activation Function	2	Sigmoid, relu
Batch Size	2	64,128,256
Epochs	3	30,50,100
Dropout	2	0.3,0.1
Learning Rate	2	0.1,0.01

Proposed model trained with the possible combinations of listed hyperparameters. For example, the number of hidden layers to an ANN could be assigned in three ways like one layer with 20 neurons, two layers with 40 and 20 neurons respectively and three layers with 40, 30 and 15 layers respectively. The proposed model resulted best Hyperparameter set with least number of epochs compared to other methos. Best result is {activation:'relu', 'batch_size': 64, 'epochs' : 30 , 'dropout' : 0.3, 'learning_rate': 0.01, 'layers':[45,30,15]} with accuracy of 97.52% and 96.58% for two-class and three-class case respectively. Table 3 and Figure 7 shows the performance metrics of the two-case proposed model. Table 4 and Figure 8 demonstrate the comparison accuracy of all reported methods in two-class and three-class models. Figure 5 and Figure 6 illustrate the performance of two-class classifier and three-class classifier in confusion matrix respectively.

Table 3: Two-Case Performance Metrics

Method	Precision	Sensitivity	Specificity	F_Score
GS	94.74	96.12	90.52	95.42
RS	93.3	97.99	88.62	95.59
BS	95.69	98.04	92.37	96.85
PSO[6]	96.65	98.54	94.02	97.58
GA[22]	94.74	98.51	90.91	96.59
ANNABC	97.13	99.02	94.87	98.07

Table 4: Accuracy Comparison Of Two-Class And Three-Class Model.

Method	Two-Class	Three-Class
GS	94.1	92.55
RS	94.41	92.86
BS	95.96	93.48
PSO [6]	96.89	95.03
GA[22]	95.65	94.72
ANNABC	97.52	96.58

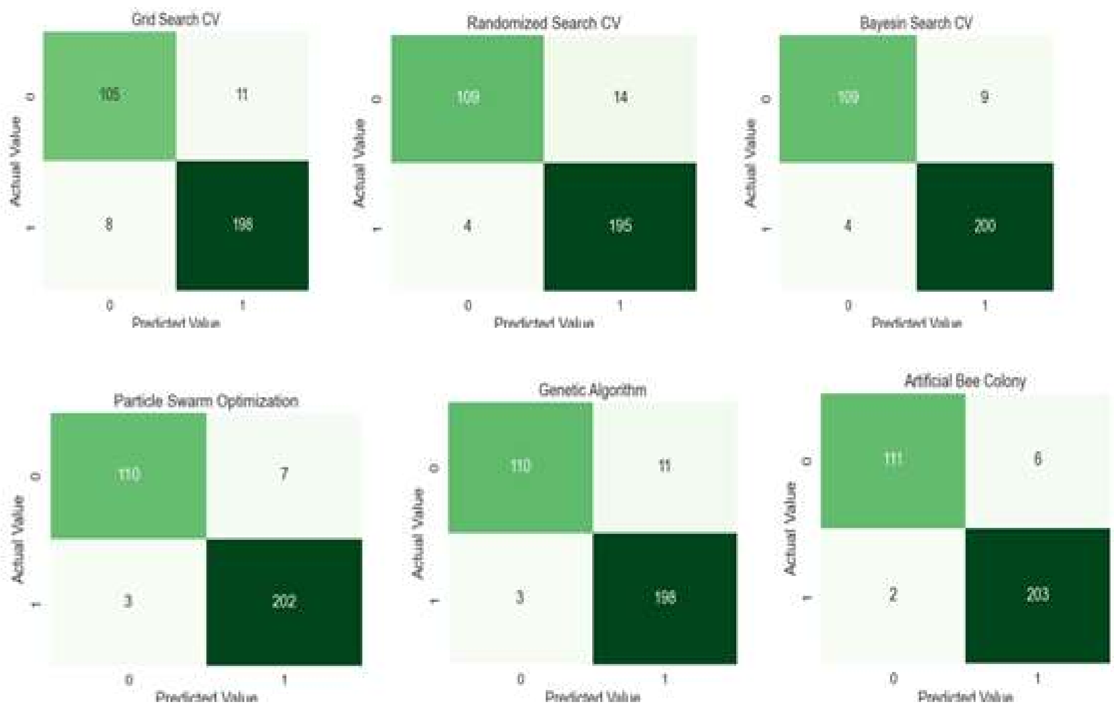

Figure 5: Two-Class Classification Confusion Matrixes A)Gridsearchcv B)Randomizesearchcv C) Bayesiansearchcv D)Paricleswarmoptimizationcv E)Genticalgorithmcv F)Artificialbeecolonycv

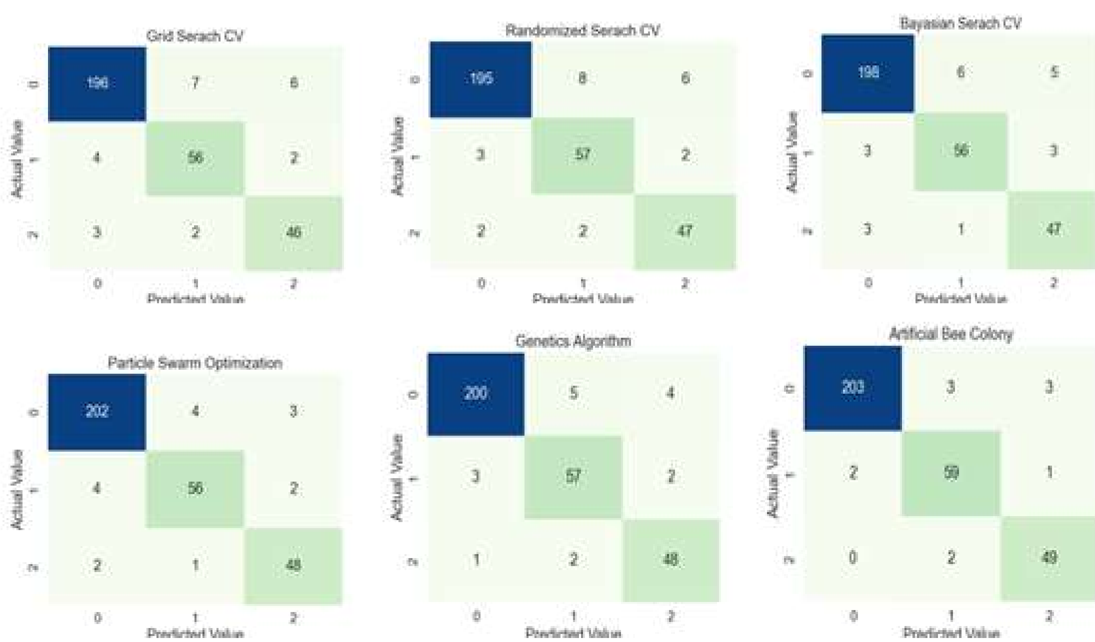

Figure 6: Three-Class Classification Confusion Matrixes A)Gridsearchcv B)Randomizesearchcv C) Bayesiansearchcv D)Paricleswarmoptimizationcv E)Genticalgorithmcv F)Artificialbeecolonycv

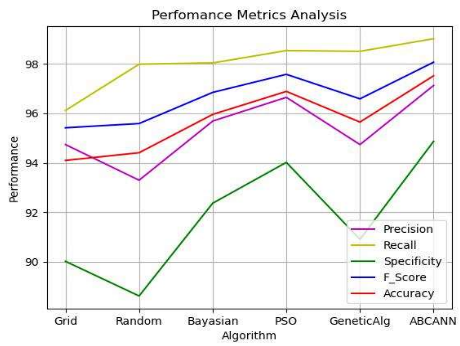

Figure 7: Performance Comparison All Methods

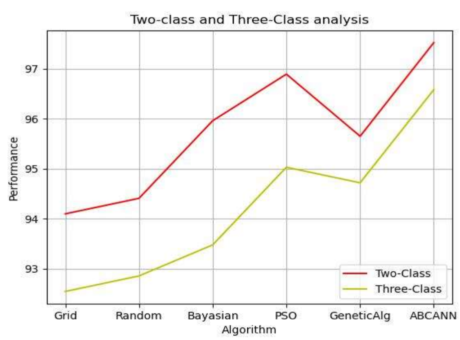

Figure 8: Two-Class Case And Three-Class Analysis

5. CONCLUSION AND FUTURE SCOPE

This study investigated at the performance of the ABC hyperparameter adjusted ANN in detecting malignancy in mammography images in the two-class and three-class models. The suggested technique produced good results: the accuracy of the two-class model was 97.52%, and the performance of the three-class model was 96.58%. Based on the findings, the categorization of the MIAS dataset provided in this work has the potential to assist radiologists in identifying breast tumours. The performance measures demonstrated that the ANN-ABC model outperformed other competing classifiers in prediction. The proposed ANN-ABC model results are displayed in a confusion matrix, which allows categorization of mammograms into "Normal" or "Abnormal" in the case of two-class and "Normal" or "Benign" or "Malignant" in the event of three-class.

In the future, we will reconnoiter ABC variants and hybrid models like ABC with other NIOA methods to fine tune the ANN hyperparameters. Finally, we will extend the ABC-based tuning process to other ML models like CNN and various breast cancer datasets.

REFERENCES:

[1] C. Li, Breast Cancer Epidemiology. New York, NY, USA: Springer, 2010.

[2] S. Nass, I. Henderson, and J. Lashof, ‘‘Mammography and beyond:Developing technologies for the early detection of breast cancer,’’ Nat.Cancer Policy Board, Inst. Med., Commission Life Stud, Nat. Res. Council, Washington, DC, USA, 2001.

[3] M. Ajay Kumar and Y.Ramadevi, ”A study: mammogram image segmentation and classification based on ABC algorithm and artificial neural networks”, Advanced Engineering Services (54/02), 2022.

[4] H. Chiroma et al., "Progress on Artificial Neural Networks for Big Data Analytics: A Survey," in IEEE Access, vol. 7, pp. 70535-70551, 2019, doi: 10.1109/ACCESS.2018.2880694.

[5] Ogundokun, R.O.; Misra, S.; Douglas, M.; Damaševičius, R.; Maskeliūnas, R. Medical Internet-of-Things Based Breast Cancer Diagnosis Using Hyperparameter-Optimized Neural Networks. Future Internet 2022, 14, 153.

[6] Vishnu, A.A., Suresh, A., Koshy, R.A., Sanjna, S., Davis, P.R. (2023). Hyper-parameter Optimised Artificial Neural Network Model for Failure Mode Identification of RC Shear Wall. In: Marano, G.C., Rahul, A.V., Antony, J., Unni Kartha, G., Kavitha, P.E., Preethi, M. (eds) Proceedings of SECON'22. SECON 2022. Lecture Notes in Civil Engineering, vol 284. Springer, Cham.

[7] Talaat, F.M., Gamel, S.A.: RL-based hyper-parameters optimisation algorithm (ROA) for convolutional neural network. J. Ambient Intell. Humaniz. Comput. (2022).

[8] J. Kennedy and R. Eberhart, “Particle swarm optimization.” Proc. IEEE International Conf. on Neural Networks (Perth, Australia), IEEE Service Center, Piscataway, NJ, 1995 (in press).

[9] L. Davis, Ed., Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, NY, 1991.

[10] Das, D., Pal, A.R., Das, A.K. et al. Nature-Inspired Optimization Algorithm-Tuned Feed-Forward and Recurrent Neural Networks Using CFD-Based Phenomenological Model-Generated Data to Model the EBW Process. Arab J Sci Eng 45, 2779–2797 (2020).

[11] Glover, Fred, and Manuel Laguna. "Tabu search." Handbook of combinatorial optimization. Springer, Boston, MA, 1998. 2093-2229.

[12] Mamindla Ajay Kumar, Dr. Y Ramadevi. Multi-Otsu's image segmentation for Mammograms using Artificial Bee Colony (ABC) Algorithm. Annals of the Romanian Society for Cell Biology, (2021),12353–12362.

[13] Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical report.Computer Engineering Department, Engineering Faculty, Erciyes University.

[14] Zhiqiang Guo, Lina Xu & Nona Ali Asgharzadeholiaee (2022) A Homogeneous Ensemble Classifier for Breast Cancer Detection Using Parameters Tuning of MLP Neural Network, Applied Artificial Intelligence, 36:1, 2031820, DOI: 10.1080/08839514.2022.2031820.

[15] Pannakkong, Warut, et al. "Hyperparameter Tuning of Machine Learning Algorithms Using Response Surface Methodology: A Case Study of ANN, SVM, and DBN." Mathematical Problems in Engineering 2022 (2022).

[16] Stephan, Punitha, et al. "A hybrid artificial bee colony with whale optimization algorithm for improved breast cancer diagnosis." Neural Computing and Applications 33.20 (2021): 13667-13691.

[17] Punitha, S., Fadi Al-Turjman, and Thompson Stephan. "An automated breast cancer diagnosis using feature selection and parameter optimization in ANN." Computers & Electrical Engineering 90 (2021): 106958.

[18] Radzi, S.F.M.; Karim, M.K.A.; Saripan, M.I.; Rahman, M.A.A.; Isa, I.N.C.; Ibahim, M.J. Hyperparameter Tuning and Pipeline Optimization via Grid Search Method and Tree-Based AutoML in Breast Cancer Prediction. J. Pers. Med. 2021, 11, 978.

[19] Erdogan Erten, Gamze, Sinem Bozkurt Keser, and Mahmut Yavuz. "Grid search optimised artificial neural network for open stope stability prediction." International Journal of Mining, Reclamation and Environment 35.8 (2021): 600-617.

[20] Itano, Fernando, Miguel Angelo de Abreu de Sousa, and Emilio Del-Moral-Hernandez. "Extending MLP ANN hyper-parameters Optimization by using Genetic Algorithm." 2018 International joint conference on neural networks (IJCNN). IEEE, 2018.

[21] P. Shunmugapriya and S. Kanmani, A HYBRID ALGORITHM USING ANT AND BEE COLONY OPTIMIZATION FOR FEATURE SELECTION AND CLASSIFICATION (AC-ABC HYBRID), Swarm and Evolutionary Computation: 2017.

[22] Ahmad, Fadzil, et al. "A GA-based feature selection and parameter optimization of an ANN in diagnosing breast cancer." Pattern Analysis and Applications 18.4 (2015): 861-870.

[23] Suckling J, Parker J, Dance D R, Astley S, Hutt I Boggis, C. R. M., Ricketts I, Stamatakis E, Cerneaz N, Kok S, L Taylor, P Betal D, And Savage J. (1994): 'The Maxnmographic Image Analysis Society Digital Maxnmograxn Database',Vol. 1069 Of Excerpta Medica International Congress Series', York, England, Pp. 375-378.

[24] N. Bhaskar and T. S. Ganashree, Lung Cancer Detection with FPCM and Watershed Segmentation Algorithms, vol. 3. Springer International Publishing, 2020.

[25]http://www.breastcancerindia.net /bc/statistics/stat_global.htm

[26]https://www.analyticsvidhya.com/blog/2021/06/tune-hyperparameters-with-gridsearchcv/

International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Energy Efficient Data Aggregation Scheme using Improved LEACH Algorithm for IoT Networks

Guguloth Ravi[1], M. Swamy Das[2], Karthik Karmakonda[3]

Submitted: 24/10/2022 **Revised:** 22/12/2022 **Accepted**: 23/01/2023

Abstract: The Internet of Things makes it possible to have connected buildings, businesses, and intelligent homes by merging embedded technology, wireless sensor networks, control and automation technologies, and wearable gadgets. It is critical to regularly monitor the energy usage of the Internet of Things network since sensor nodes have limited power. In wireless sensor networks, the most significant obstacle is the exhaustion of available energy, and extending the network's lifetime can be accomplished by lowering the amount of energy that is spent. Energy aware routing protocol is highly important in IoT-based networks, although routing protocol that simply considers energy parameter has not performed successfully in managing excessive energy consumption. Energy aware routing protocol is very important in IoT-based network. The emergence of congestion in network nodes results in an increase in the amount of energy consumed and the loss of packets. Routing algorithms should strive for energy efficiency and load balancing across diverse nodes in order to lengthen the lifetime of a network. This will allow for more nodes to participate in the network. Clustering is one of the optimal techniques for efficient data aggregation among the sensor nodes. In a clustered setup, the Internet of Things (IoT) network is partitioned into a predetermined number of smaller networks. One of the most common and widely used clustering methods is LEACH, which stands for low-energy adaptive clustering hierarchy. It is unfortunate that it has some restrictions. In this study, we suggest the use of CAW-LEACH (CONGESTION AWARE - LEACH) as a means of enhancing energy efficiency, CH stability, and the capacity to aggregate data without experiencing congestion. The enhanced protocol that has been proposed takes into account both the depletion energy ratio (DE) and the expected remaining energy (PRE) of the nodes while selecting CH and generating random numbers. Its purpose is to ensure that the CH node that was just recently elected will not be given a second chance in this round. This technique establishes a correlation between the threshold that is utilised in conventional LEACH and each node's energy consumption ratio. The proposed congestion aware data aggregation scheme aggregates the data through traffic free paths by estimating the congestion indicator (CIN) & link error rate (LER), Residual Energy (RE) of the all-available routing paths. By comparing the suggested method to other energy-efficient data aggregation schemes, according to the findings of the experiments, the proposed technique increases the network's durability.

Keywords: *WSN, energy consumption, data aggregation, LEACH, link error rate, congestion indicator, clustering, CAW-LEACH.*

1. Introduction

Many IoT applications also provide access to highly developed software and communication services, in addition to Internet connectivity. The Internet of Things (IoT) is a network of devices, appliances, and other items that were not previously connected to one another. There are many examples of where IoT has been put to use, including in projects that aim to boost computer and network efficiency [1]. Self-configuring wireless networks connect everything that can be connected in a wireless network. There is something in this network that serves as a communication link. In their present form, these open lines provide a wide range of possible methods of contact. Radio-frequency identification chips are the backbone of the Internet of Things (RFID). A Wi-Fi layer, which can be found at the internet's very core, is being used to build out the worldwide infrastructure for RFID tags. The network allows for the exchange of information between the various devices and computers that are linked to one another. These components are a part of a larger, more intricate system. Data is gathered from a variety of sensors to aid in taking temperature readings and other measurements in the immediate vicinity of the sensor. Data is transmitted to nearby sensors so it can be analysed and interpreted in accordance with the requirements of the currently active applications [2].

Many different routing techniques are implemented in order to cut down on the amount of data transfer and, as a result, the required amount of power [3]. When using a WSN, data are transmitted from the sensor node to the

[1]*g.raviraja@gmail.com, Research Scholar, Department of Computer Science and Engineering, University College of Engineering (UCE), Osmania University (OU) ,Asst.professor in MRCET,Hyderabad,Telangana,India.*
[2]*msdas_cse@cbit.ac.in, Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Hyderabad,Telangana,India.*
[3]*Karthikk_cse@cbit.ac.in, Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Hyderabad,Telangana,India.*

base station, which is often situated in the geographic centre of the network. If the sensor node is placed a significant distance from the base station, then the data packet will need to travel a long distance, which will result in a higher amount of power consumption. In addition, the sensor that is placed near the base station, which is referred to as the bottle neck zone, is subject to significant traffic flow, which quickly depletes the energy stored in the node's battery. Therefore, the nodes in the bottleneck zone die off more quickly, which has an impact on the longevity of the network as well as its connectivity [4].

Routing protocols that simply take energy into consideration as one of their parameters are inefficient. Utilizing a wide variety of parameters not only helps reduce energy consumption but also makes routing protocol more effective [5]. When dealing with various applications, one must take into account a variety of aspects. Management of congestion is considered to be one of the most critical aspects. The network uses more energy when congestion sets in, which is a negative energy-cycle outcome [6]. Congestion can arise in networks for a variety of different reasons. A lack of available storage capacity in the nodes that make up the relay network is one of the primary causes. Congestion happens if a node receives more packets than it can handle, and when this happens, many of the received packets are lost. In wireless sensor networks, congestion developed for almost the same reasons as in wired sensor networks. Congestion can develop, for instance, when multiple nodes all at once make the decision to send packets via a common media [7].

The practise of clustering is one of the promising and effective strategies that can be used to improve energy efficiency [8]. The construction of clusters and the use of a variety of communication channels for the purpose of data transmission have been the approaches that have received the most attention. Figure 1 presents a schematic representation of the conventional clustered network topology.

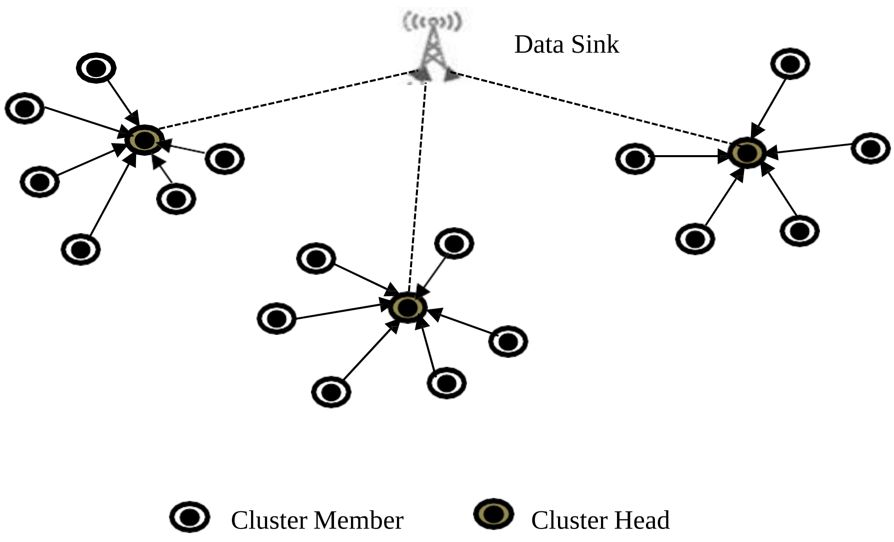

Fig 1: Clustered WSN network topology

Cluster-based routing techniques, in comparison to non-clustering routing protocols, are able to make more effective use of the sensor nodes that are present in the network. One of the responsibilities of a cluster leader, also known as the cluster head (CH), is the elimination of correlated data, which can result in a reduction in the total volume of data [9]. After that, the CH will send the BS the accumulated data that it has just finished processing. Cluster-based routing protocols separate sensor nodes into a large number of clusters with the goal of reducing the amount of energy needed for long-distance communication. Due to the considerable difference in energy depletion between CHs and other nodes, clustering can help reduce the overall amount of energy that is consumed and maintain a workload that is equitable across all of the nodes. Therefore, clustering is a solution that is good for the environment and energy efficiency, as it extends the lifespan of networks and improves energy efficiency. In addition, the vast majority of clustering algorithms make use of optimal CH selection in order to forestall the untimely demise of the sensor nodes and further extend the network's lifetime [10].

Low Energy Adaptive Clustering Hierarchy Protocol

According to [11], the LEACH protocol was the first clustering-based routing method to provide scalability and extend the life of a network. LEACH allows for a notable drop in worldwide energy consumption by continuously dispersing network load to all nodes in varied regions. In a typical setup, clusters are used to organise sensor nodes hierarchically, with one node serving as the CH for its cluster. As the group's central hub (CH), your task is to gather information from the other nodes in your cluster, organise it into reports, and send them off to the sink node. The LEACH algorithm selects a node as the CH if its probability, given by a random number between 0 and 1, is less than a certain threshold, denoted by the symbol TH(n) (Eq 1). In order to join a certain cluster, the remaining nodes pick the CH that requires the least amount of communication energy to reach. By doing so, they are able to join the rest of the cluster. CH's job include switching between all of the sensors to spread out the battery drain.

$$TH(n) = \begin{cases} \frac{P}{1-P\left[\left(r \bmod \frac{1}{P}\right)\right]}, & if\ n \in G \\ 0, & otherwise \end{cases} \qquad \text{Eq (1)}$$

In the equation that was just shown, $TH(n)$stands for the threshold value of n nodes, and P is the abbreviation for the probability value. Every node in the network selects a random integer between 0 and 1, starting with 0.

LEACH operates in iterative rounds, with the first phase of each round being the setup phase, which is in charge of clustering the nodes together [12]. After this, a steady-state phase occurs, during which data is sent to the sink node. In order to cut down on unnecessary overhead, the steady-state phase is significantly longer than the setup phase. Once a node has reached the conclusion that it will function as a CH during the setup phase, it will proceed to transmit an advertisement message. After receiving this message, every node that is not already part of a cluster will make the choice to become part of one of the CHs. Each participating node must participate in data transmission during the steady-state phase. The CH compiles data from all of the cluster nodes and transmits it to the final destination sink. Because of its decentralised nature, LEACH doesn't need any kind of control data from the sink. Furthermore, LEACH can function without any nodes being pre-aware of the global network.

The LEACH protocol makes it possible to keep the network running for longer. On the other hand, there are significant flaws in the new LEACH-based protocols that need to be fixed before they can be considered truly effective. The following are a few of these issues, as well as the answers to them.

- LEACH makes the assumption that all nodes are capable of communicating with the sink, which limits the scalability of the protocol. Incorporating multi-level clusters and providing support for multi-hop routing are two potential solutions to this problem.
- The overhead that is involved, which is caused by fluctuations in CH, results in an inefficient use of energy. One solution to this issue is to cut down on the number of rounds that must be completed during the cluster rebuilding phase.
- The likelihood of picking CHs does not take into account the energy that is still present in the nodes. As a result, nodes that have a low amount of energy left over may be picked as CHs, which results in a rapid loss of energy for these nodes and ultimately leads to the disconnection of the entire cluster.
- The total number of CHs is subject to significant swings. This factor results in uneven cluster partitions, which in turn leads to an increase in the overall amount of energy that is lost over the entire network.

In this paper, we present a unique improved congestion aware LEACH (CAW-LEACH), with the goal of overcoming the shortcomings of traditional approaches and further extending the lifetime of IoT based networks. Each node's depleted energy ratio is taken into account in conjunction with the threshold used in conventional LEACH. This ensures that the node that was previously elected CH will not have another chance to win in the current round. This paper proposes a hierarchical data aggregation scheme as a congestion aware data aggregation scheme to improve energy efficiency and prolong the lifespan of the network. By estimating the congestion indicator (CIN), link error rate (LER), and residual energy (RE) of all available routing paths, the proposed congestion aware data aggregation scheme is able to aggregate the data via traffic-free paths.

Paper contributions

- The threshold and random number generation are related to the depleted energy ratio (DE) of the sensor nodes in order to maintain an appropriate level of energy consumption and to lengthen the lifetime of the network.
- The CHs who were chosen in the round before will not be chosen in the round after that since their DE is significantly higher in comparison to the nodes that are not CHs. It eliminates the possibility that the CHs node that was previously elected will get a second opportunity in following rounds.
- The proposed energy efficient and congestion aware data aggregation determines the routing path for data aggregation with the lowest cost subjected to end to-end delay and congestion.

2. Literature Survey

When designing WSNs, energy conservation is a critical factor to consider. The life of the battery is completely dependent on the longevity of the network. When trying to extend the lifetime of a WSN, one of the tactics that can be implemented is called clustering.

A fresh approach to the problem of constructing for HWSN, an energy-efficient clustering technique has been developed proposed by Santhosh V. Purkar and colleagues [13]. Several parameters, such as residual energy, starting energy, and hot count, as well as the CH selection, are taken into account during the clustering process. Energy efficiency, lifetime, stability, and throughput of the HWSN protocol are improved when using the proposed method as compared to using SEP, DEEC, or LEACH.

Hybrid unequal energy-efficient clustering and layered multi-hop routing was developed by Seyed Mostafa Bozorgi et al. [14] for use in WSNs. Based on the information about the neighbouring nodes in the clustering, a protocol method is defined, and layering is performed. Power consumption and network congestion are both reduced when unused control messages are purged. Based on simulation results, the newly developed HEEC technique is superior to the FHRP, LEACH-ERE, EADUC-II, and HUCL for achieving higher levels of stability.

A method for an energy-efficient clustering approach that utilises an optimised version of the LEACH protocol for data collection and transmission was provided by Salil Bharany et al., [15]. The approach that has been presented for data transmission uses less energy than other methods already in use. When compared to the currently utilised LEACH protocols, the proposed solution yields superior performance outcomes in terms of the ratio of packets successfully delivered and the amount of energy saved.

Sadia Firdous and colleagues [16] have presented a clustering-based routing strategy for optimising energy resources while taking load balancing into consideration as part of their research. The rotation of the cluster heads and the calculation of the distance have been handled, together with the energy utilisation at the sensor nodes. In comparison to the LEACH algorithm, the simulation's findings demonstrate that the suggested approach works offers significantly increased performance in terms of both the average amount of energy consumed and the lifetime of the network.

The layered architecture and the method for balancing load between CHs have been presented by Salim El Khediri et al., [17] in order to facilitate the handling of data packets. When it comes to the routing of data packets across sensor nodes, the network is divided into sections of unequal sizes. The new protocol offers significantly enhanced functionality in terms of energy usage, network longevity, and dependability.

In their presentation, Waz et al. [18] covered a wide range of Internet of Things (IoT) security products and approaches. When dealing with many IoT platforms, all of the components are combined into a single stack. This allows the integrity of the components to be maintained. This integration from one stage to another guarantees that there will be no break in the continuity of the security. With the assistance of middleware, complete data from the Internet of Things device may be tracked back user, and the user to the system.

Israr Ahmed et al. [19] described the most significant implications that the Internet of Things has on our everyday lives. It is presumed to be the system that has widespread adoption in both the virtual and the real world. As a result, this article delves into all of the most pressing concerns regarding confidentiality and safety.

According to the results that were obtained [20], this strategy is capable of handling the issues with energy efficiency as well as data delivery-related issues that arise inside heterogeneous systems. The results of the simulation reveal that the proposed method is improved and is superior to the existing ways. These findings are determined by the class's methods. Following the conclusion of each round, the energy consumption models are formed and subsequently decomposed into their respective gates. All in all, this extends the network's lifespan.

Using an enhanced version of the LEACH protocol, Salem and Shudifat [21] looked at the power sensitivity involved in selecting normal nodes as cluster heads when they are closer to the BS.Because of the shortcomings of the LEACH methodology, this was done. The results of the simulations show that the power consumption can be decreased while simultaneously increasing the network lifetime and maximising the number of cycles. In comparison to the LEACH protocol, the proposed method both reduces energy consumption and increases network longevity.

AnshuKumar Dwivedi and colleagues [22] have showed how to get over the bottlenecks that exist in WSNs by using balanced energy dissipation. Fuzzy inference algorithms are used in the clustering process and CH is chosen from those clusters have been studied by the authors. The findings of the experiments demonstrated that the recently proposed method known as EE-LEACH provides superior performance when compared to the older procedures known as DFCR and SCHFTL. The summary of research deficiencies is presented in Table 1.

Table 1: Summary of research gaps

S No	Year	Methodology	Techniques used	Parameter improved	Research gaps
1	2018	Energy efficient Clustering protocol to enhance performance of Heterogeneous wireless sensor network	HWSN, EECP	The node quality index, the length of the stability period, the number of living nodes in each cluster round, and the number of dead nodes in each cluster round	The best CH role is not determined in an appropriate manner. The mobile sink technique was not utilized in the planned
2	2019	Hybrid unequal energy efficient clustering	HEEC	Extended Network Lifespan, Lower Network Overhead, Greater Network Stability, Better Energy Balance	For the purposes of routing, these nodes are not considered energy harvesting nodes. There is no assurance that the network will save energy or have a long lifespan.
3	2021	Energy efficient LEACH	LEACH-EE	Lifespan, amount of energy left over, and packet success ratio	It is necessary to enhance the life cycle of WSNs and reduce the amount of energy required for data transmission.
4	2021	Distance Energy Evaluated approach	DEE	Lifespan of the network, percentage of successfully delivered packets, and number of active node	This work is limited by how long it will last. We have measured how long the node will last because we don't have a lot of resources.
5	2019	Enhanced LEACH	LEACH, CH Selection, E-LEACH	Energy usage at the cluster's head nodes and energy consumption across the network as a whole	This approach does not include any energy-efficient relay selection options. The phase that deals with the transfer of data has a very high energy usage.
6	2021	Fuzzy-LEACH	CH selection, fuzzy based clustering	Enhanced the network's average amount of energy. The mount of throughput during the stable time.	The greater the distance, the higher the energy levels. The data aggregation step has not been modified in accordance with the clusters.

3. System Model

Energy usage model

A sensor node's power consumption is a major factor in establishing the maximum level of performance that can be attained by the system. The network protocols that are put in place are constantly working to lower the amount of energy used by the system in an effort to prolong its lifespan. The energy consumption model is depicted in Figure 2.

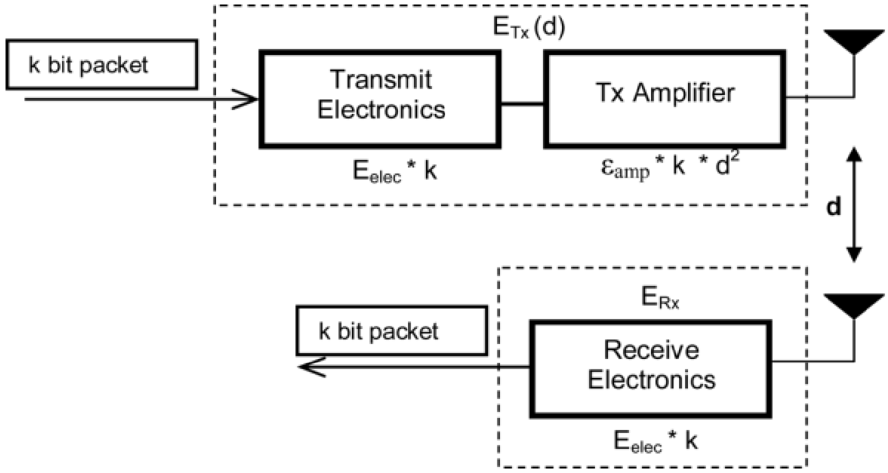

Fig 2: energy usage model in WSN

Radio waves are used for inter-node communication, and an energy model is applied to calculate how much electricity will be expended during the transmission and reception of data. The energy model takes into account not only the free space channel but also the multi-path fading channel. The free space channel is used if the distance (d)between the transmitter and receiver is less than a threshold value, called d_0, and the multi-path fading channel is used otherwise. Simply plug your values into the following equation to find out how much energy the model predicts is needed to send a k-bit message over a d -mile distance.

$$E_T(k,d) = \{ \begin{matrix} kE_{elec} + k \in_{fs} d^2, \ d \leq d_0 \\ kE_{elec} + k \in_{mp} d^4, \ \ d \geq d_0 \end{matrix}$$

where d_0 represents a threshold, E_{elec} represents the amount of energy needed by an electronic circuit, $\in_{fs}$represents the amount of energy needed by free space, and $\in_{mp}$ represents the amount of energy needed by a multi-path channel. The formula for calculating the amount of energy needed to receive k-bits of data is as follows:

$$E_R(k) = kE_{elec}$$

The amount of energy that is used by an electronic circuit, denoted by the symbol (E_{elec}), is determined by a number of different parameters. These factors include the digital coding, modulation, filtering, and spreading of the signal, among others. Both the distance between the transmitter and receiver as well as the bit error rate affect the amount of energy that is used by the amplifier whether operating in free space ($\in_{fs}* \ d^2$) or when operating in multi-path($\in_{mp}* d^4$).

Proposed system

Proposed CAW-LEACH method

Here we discuss the proposed approach for CH selection and how it stabilises random number generation are broken down and discussed in detail. In the method that has been proposed, the generation of random numbers is multiplied by the depleted energy (DE) and expected remaining energy (PRE) parameters of the network. This is done so that the algorithm is dependent on the energy that is contained within the nodes.

The first step in the CH selection process used by the traditional LEACH algorithm is for the nodes to generate random numbers. After that, the resulting random number is compared to a predetermined cutoff value. In the event that the random value is lower than the threshold, the node will be elevated to CH for the duration of the round. If the random number is less than the cutoff, then this holds true. While there are advantages to using the LEACH method, it cannot guarantee the CH's current residual energy. Standard LEACH-based protocols choose the threshold $TH(n)$ without regard to the nodes' energy levels, instead considering only whether or not the nodes will be selected as the CH. For this reason, CHs are selected at random, and if a node with lower energy is assigned as CH, it will soon die.

The suggested method makes the selection of CH reliant on the depleted energy (DE) and expected remaining energy (PRE) factors, which ultimately results in an increase in the system's performance as well as the network's longevity. The production of the random number is altered in the way that has been proposed to make the selection of CH more energy efficient. The random number is then multiplied by the sensor nodes' depletion energy (DE) and expected remaining energy (PRE) values. As a consequence of the product of these components being multiplied together, the method for producing random numbers is now reliant upon the energy of the nodes. The following is an explanation of the parameters for the suggested depleted energy (DE) and expected remaining energy (PRE):

Depleted energy (DE) ratio: The parameter depleted energy ratio can be defined as the ratio of energy depleted in the sensor rounds in the previous rounds. This parameter is very important to evaluate to prevent the same node is elected as CHs in the subsequent rounds. The parameter DE can be mathematically expressed as follows:

$$DE = \frac{E^T_{init} - E^T_{res}}{(r-1)_T}$$

Where E^T_{init} denotes the starting energy, E^T_{res} denotes the residual energy of each node, r denotes the current round, and $r - 1$is the round that came before this one at time T. The performance of DE in the previous round will serve as the criterion for choosing CH in the next round. In the subsequent round, the role of CH will go to the node that finished the previous round with the lowest DE. Because a CH from the previous round has an extremely high DE in compared to other nodes that are not CHs, it will not be chosen as a CH in the following round.

Predicted remaining energy (PRE): The difference between the initial energy, which is denoted by E_{init}, and the entire energy, which is denoted by E_{ec} , is what constitutes the remaining energy, which is denoted by E_{pre}. Because the prediction of energy avoids participating in CH candidate networks with weaker nodes, E pre is evaluated in this situation rather than residual energy (RE).

The predicted remaining energy of node "n" is given by

$$E^T_{pre}(n) = E^T_{init}(n) - E^T_{ec}(n)$$

Where, E_{ec} of node n can be calculated as follows,

$$E_{ec}(n) = E_{i,n} + E_{n2BS} + E_{elec} + E_{DA}$$

Here, $E_{i,j}$&$E_{node2BS}$ are the transmitting energy for "l"bits from node "i" to "n" and "n"to base station or sink, respectively. Additionally, E_{DA} is the aggregating energy of a datum, and E_{elec} is the energy spent by the reception circuit per bit.

Rand is the symbol used to represent the typical random number in LEACH's classic $rand\ (n)$. In the method that has been suggested, the enhanced procedure for selecting a random number can be described as follows, according to the Equation:

$$rand(n)' = rand(n) * (DE_n + PRE_n)$$

As the enhanced random number's value has been calculated, it can be checked against the sensor nodes' thresholds. When choosing a CH, the threshold function should be given careful thought. The threshold function evaluates the node probabilities and uses that information to select the CH. By judiciously factoring in node energy to the threshold function, network performance can be fine-tuned. In order to keep the network's power consumption stable, each node performs some CH operations. Based on the likelihood of CH selection, there is a one in one possibility of any one of the potential nodes in the network being chosen as the CH. A node's ability to function as a CH in the network is dependent on how much energy it has. The suggested method uses the energy of the nodes from the very first node in the network all the way to the very last node to perform the threshold function.

At this stage, the value of the manipulated random number is evaluated in relation to the threshold function.

$$rand(n)' \leq TH(n)$$

Where $TH(n)$can be represented as:

$$TH(n) = \begin{cases} \frac{P_n}{1 - P_n\ [(r\ mod\ \frac{1}{P_n})]} * (DE + PRE) + \frac{r}{P_n}, & if\ n \in G \\ 0, & otherwise \end{cases}$$

In the equation that was just presented, P_n stands for the chance that n will become CH, and r stands for the round number.

Node is promoted to CH status if the random number is less than the threshold; otherwise, the process moves on to the next node in the list.

After the CH selection process is over, the CHs will spread the news to the other nodes in the network that they have been chosen to serve as the CH for this iteration. In order to successfully finish this operation, Each CH node will notify every other node in the network via a broadcast message disguised as an advertising. The strength of the signal of the message that is sent from the CH nodes is used by each member node to determine whether or not it will take part in the process of cluster building.

Proposed Congestion aware intra-cluster routing

In this research, we propose a congestion aware data aggregation technique that is based on congestion indicator (CIN), link error rate (LER), and residual energy (RE) characteristics. Our goal is to achieve more reliable data transmission using this scheme. In order to design the many feasible routing paths from source nodes to CHs, the congestion indicator and the link error rate of all of the available links are calculated for each cycle of communication.

*Congestion indicator (CIN):*Congestion happens when the number of packets arriving at a given node is greater than the number of packets that it can forward. This leads to an increase in the rate at which packets are dropped and a delay in their delivery. The time that elapses between two consecutive packet arrival times at a given node is referred to as the packet arrival time T_{PA} . The packet arrival rate R_{PA} decreases with increasing delay. Inversely proportional is the ratio of packet service time ST_P to packet service rate SR_P. The service time of a packet is the amount of time it takes

delay caused by waiting to forward packets in the queue or for a packet to be retransmitted.

Following are the numbers for the packet arrival rate and the packet service rate:

$$R_{PA} = {}^{1}/_{T_{PA}}$$

The congestion indicator (also known as CIN) is helpful in locating instances of congestion. The ratio of the rate at which packets arrive to the rate at which they are serviced is used to measure it.

$$CIN = \frac{R_{PA}}{({}^{1}/_{S_{TP}})}$$

If $R_{PA} > SR_P$, then the CIN value is greater, which indicates that the node has been congested. The higher the CIN value, the greater the amount of congestion that will result at the particular node. The rate of the packet service that was utilised in order to identify the congestion.

If the CIN number is lower, then a node will experience a lower level of congestion. The following formula is used to compute the cost of the route based on the traffic index heading towards CH:

$$CIN_{route} = \sum_{i=1}^{n} CIN_n$$

Link Error rate (LER):The error rate (ER), which has a negative impact on the functioning of the sensor network, is one of the issues that have traditionally been linked with wireless channels. The linkages that have a high mistake rate will result in an increased cost to route, and will therefore be avoided. It is sufficient, then, to take into account the link error rate in order to achieve an improvement in the energy efficiency of the data aggregation process. The following is a description of how the parameter LER between nodes i and j can be understood:

$$LER = \frac{dist(i,j)}{Buffer_{size}(j)}$$

Data transmission phase

The CH considers both the congestion indicator and the link error rate characteristics while making its assessment of all of the available links. Following an analysis of the optimal paths, the cluster head generates a routing table for every node in the cluster. When determining the most efficient path, the routing table takes into account a unique record. The following fields can be found in the routing table:

Id	Energ	CIN	LER

Cluster head sends routing tables to each node after constructing the routing table. All nodes have routing tables at the end of the process. The primary objective of this phase is to identify the subsequent hop for each packet that arrives at each node. Once a data packet is received, the receiving node act as follows: If the next hop node is CH, the data should be transmitted directly to CH. Else, check for the available neighbours. If neighbours found, then compare the CIN of each neighbour and select the node with lowest CIIN & LER value as forwarder node and forward the data to the selected node. If all the neighbour nodes having similar CIN value, then select the node with LER and transmit the data.

Energy analysis

Let's say that the entire number of sensor nodes that are present in the whole sensor region is denoted by N. Because there are k total sensors in the cluster that was generated, the total number of clusters in the separated region is equal to N/k. The M sensor nodes each send out the same number of bits of data. According to the Radio Energy Dissipation model of the sensor node, the amount of energy that is needed to transmit and receive 'l' bits of data over a distance of D_n is denoted by the symbols $E_{trans}(l,d)$ and $E_{recv}(l,d)$ accordingly, such that

$$E_{trans}(l,d) = \begin{cases} lE_{ec} + l\varepsilon_{fs}d^2, & d \le d_0 \\ lE_{ec} + l\varepsilon_{mp}d^4, & d \ge d_0 \end{cases}$$

$$E_{recv}(l,d) = lE_{ec}$$

The amount of energy that must be used during transmission is proportional to the path loss exponent n, where n equals 2 for open space and 4 for multipath interference. E_{ec} is the amount of energy that is used up by the electronics of the transmitter and receiver during a single bit.'d' denotes the physical separation between the transmitter and the receiver. The amount of energy that is used by the transmitter amplifier may be calculated using the route loss exponent and the formula ε_{fs}, ε_{mp}. Due to the fact that the sensor nodes do aggregation, the'm' bits of data are aggregated into a smaller number of the 'h' bits. The value of E_{CD} represents the amount of energy that is used by the model that utilises centralised data aggregation (also known as Cluster Head). The value given by E_{SD} represents the amount of energy that is used by the suggested approach for performing Data Aggregation in the sensor node. The energy gain, denoted by E_{gain} , for a cluster that is a result of the suggested technique is as follows:

$$E_{SD} = m(E_{ec} + \varepsilon_{fs}d^2) + mE_{ec}$$

$$E_{CD} = h(E_{ec} + \varepsilon_{fs}d^2) + hE_{ec}$$

$$E_{gain} = E_{SD} - E_{CD}$$

$$E_{gain} = (2m - h)E_{ec} + (m - h)\varepsilon_{fs}d^2$$

Because there are N/k clusters in the area surrounding the sensor. The total amount of energy that can be gained via the proposed method is denoted by b.

$$Total_{gain} = (\frac{N}{K}) E_{gain}$$

Algorithm

For all the nodes 'n' where $n \in N$

Divide the nodes as 'k' clusters

End for

CH selection

For all node 'n' where $n \in k$

Calculate CER_n

Calculate $TH(n)$

Estimate $rand(n)$

$rand(n_{new}) = rand(n) * CER_n$

If $rand\ (n_{new}) \leq TH(n)$

$CH -> n;$

End if

End for

Data transmission

For all the available paths

Calculate CIN, LER

If $(CIN_n\ \&\ LER_n = low)$

Select n as forwarder node

Else

$n = n + 1$

End if

End for

4. Result and Discussion

Setup for simulation

The newly proposed CAW-LEACH mechanism is simulated, and its performance is evaluated with the help of NS2 before being compared to EN-LEACH and EE-LEACH. The sensor nodes in the network field are placed there in a haphazard manner. The dimensions of the network area are 1000 metres by 500 metres. All of the sensor nodes have been configured to have an initial energy level of 100 j. The number of nodes in the network can range anywhere from 50 to 200. During data transmission, the CBR traffic agent is utilised to generate consistent traffic. CBR is an abbreviation for "constant bit rate." UDP is the protocol that is used to carry out the data communication. The results of the experiment are presented in table 2, which can be found down below.

Table 2: Simulation parameters

Simulation Parameter	**Value**
Area of network	1000 m x 500 m
Number of sensors	50 to 200
Size of clusters	4
Initial energy of sensor node	100j
Size of packet	1024 bytes
Routing protocol	AODV

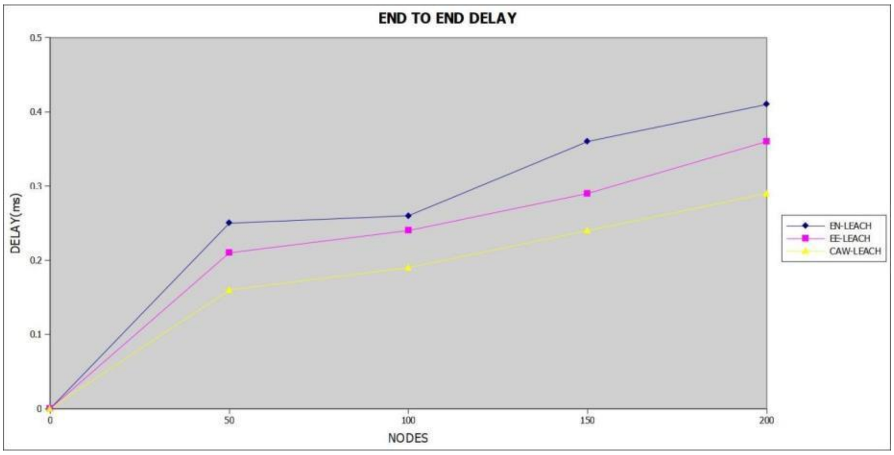

Fig 3: End to End Delay

Table 3: Comparison analysis of CAW-LEACH and existing methods as EN-LEACH and EE-LEACH for Delay

NODE	PROPOSED	EE-LEACH	EN-LEACH
50	0.16	0.21	0.25
100	0.19	0.24	0.26
150	0.24	0.29	0.36
200	0.29	0.36	0.41

The end-to-end delay is the overall time a packet takes to travel through a network. The evaluation of the proposed method's end-to-end delay time is presented in the values that have been mentioned above in table 3. The reduced amount of CH rotation is due to the stable CH selection, the implementation of DE, and the random number. The selection of low overhead relays during the relay selection process, which is conducted utilising quality and energy are connected in the QoS reputation paradigm, is another factor that contributes to the proposed method's ability to reduce end-to-end delay. When compared to the suggested technique, the previous methods encountered significantly higher delays of up to 0.48 milliseconds, while the proposed method only suffered a minimum average delay of 0.16 milliseconds. The graphical representation of the end-to-end delay can be found in Figure 3.

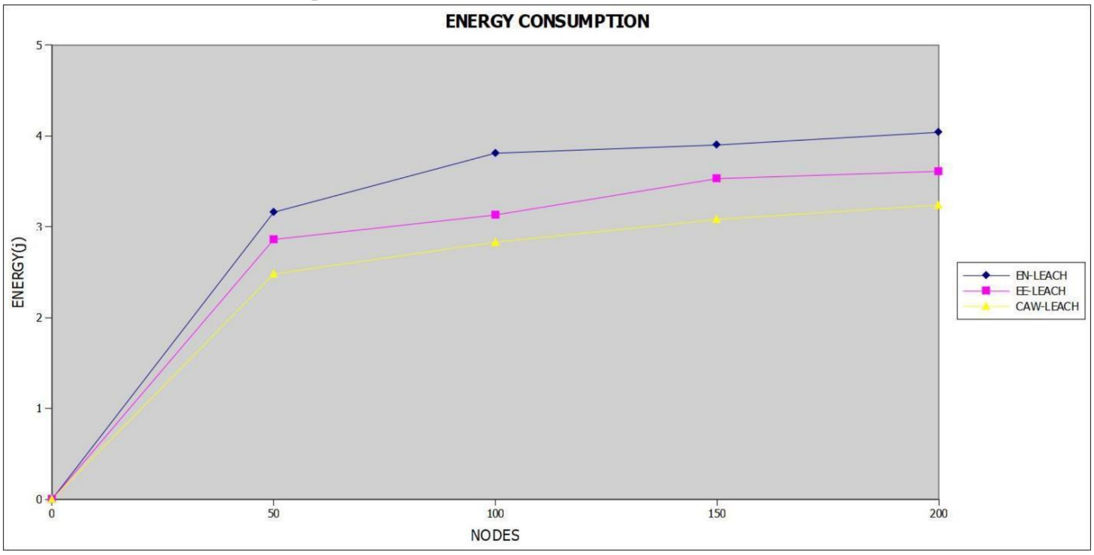

Fig 4: Energy consumption

Table 4: Comparing proposed and existing energy consumption strategies

NODE	PROPOSED	EE-LEACH	EN-LEACH
50	2.48	2.86	3.16
100	2.83	3.13	3.81
150	3.08	3.53	3.90
200	3.24	3.61	4.04

In order to participate in network operations, each sensor node is provided with 100 j of initial energy to use. Every time there is activity on the network, the energy level drops. In every network, the energy should be optimised for maximum efficiency in order to sustain network activity. The elimination of the need for retransmission and other activities that use up a lot of energy in the network is made possible thanks to the CH selection using DE and the selection of relay nodes with less congestion indicator values. This leads to a low amount of energy consumption in the proposed network. The method that was suggested had a documented rate of energy consumption that was an average of 2.9j. The graphic representation of the amount of energy used is shown in Figure 4. The findings of the experiment are presented in table 4, which can be found up above.

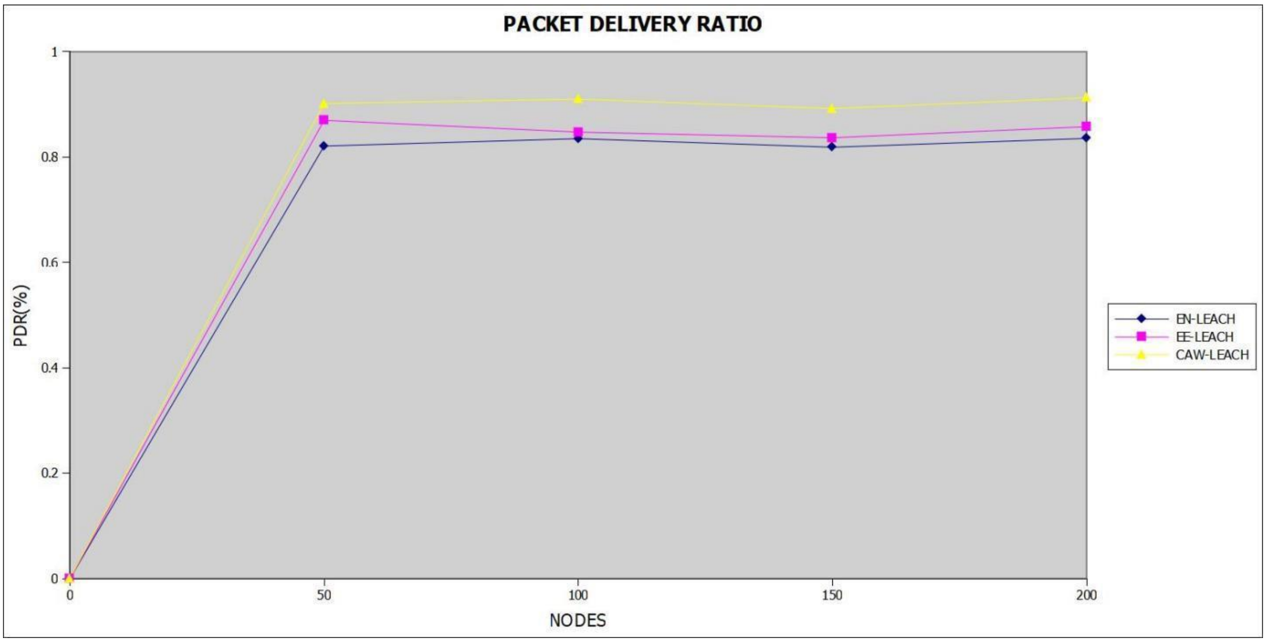

Fig 5: Packet delivery ratio

Table 5: Analysis of proposed and current PDR approaches in comparison

NODE	PROPOSED	EE-LEACH	EN-LEACH
50	0.9015	0.8696	0.8208
100	0.9099	0.8472	0.8348
150	0.8918	0.8365	0.8185
200	0.9125	0.8575	0.8353

The packet delivery rate (PDR) is defined as the fraction of data packets that reach their destination out of the total number of packets delivered. This ratio can be thought of as the proportion of data packets that are successfully received. The delivery rate was significantly enhanced because to the aggregation of data and efficient relay selection. The sensor nodes are able to convey the data with a high level of success thanks to the careful choice of channels that are reliable and well-balanced, in addition to the most suitable relays based on quality of service reputation. The prior approaches maintained an average PDR rate of 0.82, which is a rather low PDR rate. In contrast, the proposed method reached a maximum PDR of 0.91%, whereas the existing methods only achieved a PDR rate of 0.82 on average. The graphical representation of the packet delivery ratio can be seen in Figure 5. The outcomes of the experiment are presented in table 5, which can be found up above.

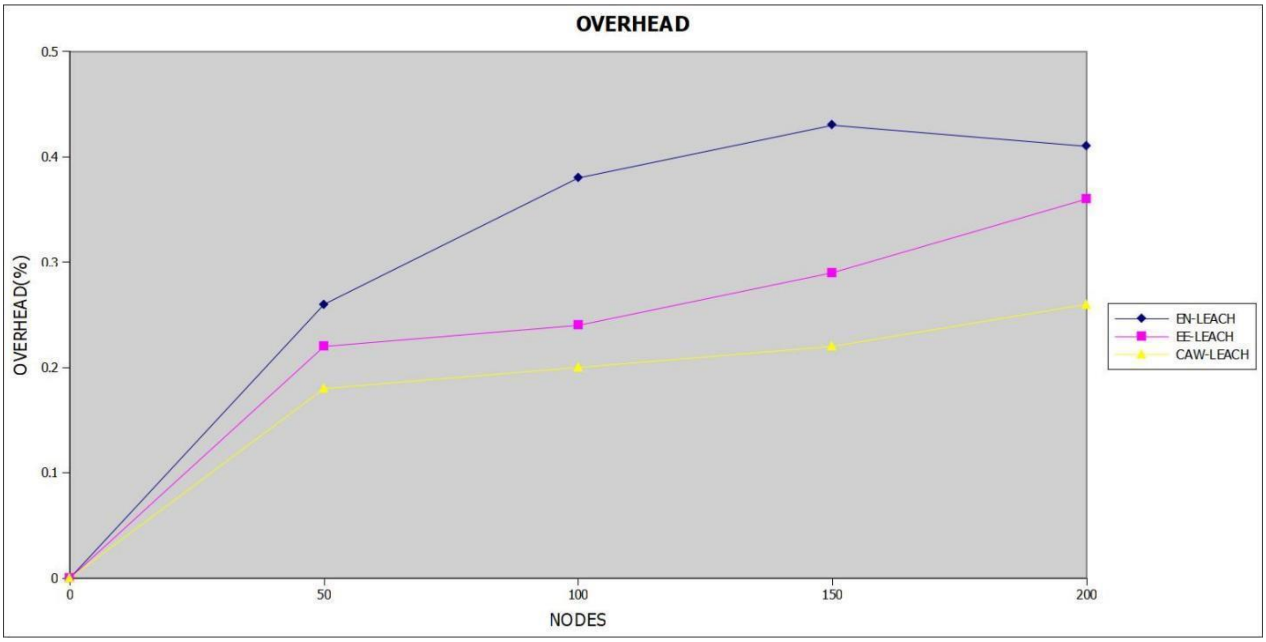

Fig 6: Routing overhead

Table 6: An evaluation of the proposed and existing approaches to Overhead

NODE	PROPOSED	EE-LEACH	EN-LEACH
50	0.18	0.22	0.26
100	0.2	0.24	0.38
150	0.22	0.29	0.43
200	0.26	0.36	0.41

The amount of control packets that are disseminated throughout the network to facilitate data transfer is directly connected to the overhead. The proposed approach had an overhead that ranged from 0.21 to 0.26 on average, whereas the existing methods had overheads that ranged from 0.26 to 0.40 on average. The error-free operation of the specified pathways is ensured by the utilisation of optimal relay selection in conjunction with congestion aware relay selection. Because of this, the proposed technique has a relatively minimal amount of overhead. The graphical representation of the routing overhead is provided in Figure 6. The findings of the experiment are presented in table 6, which can be found up above.

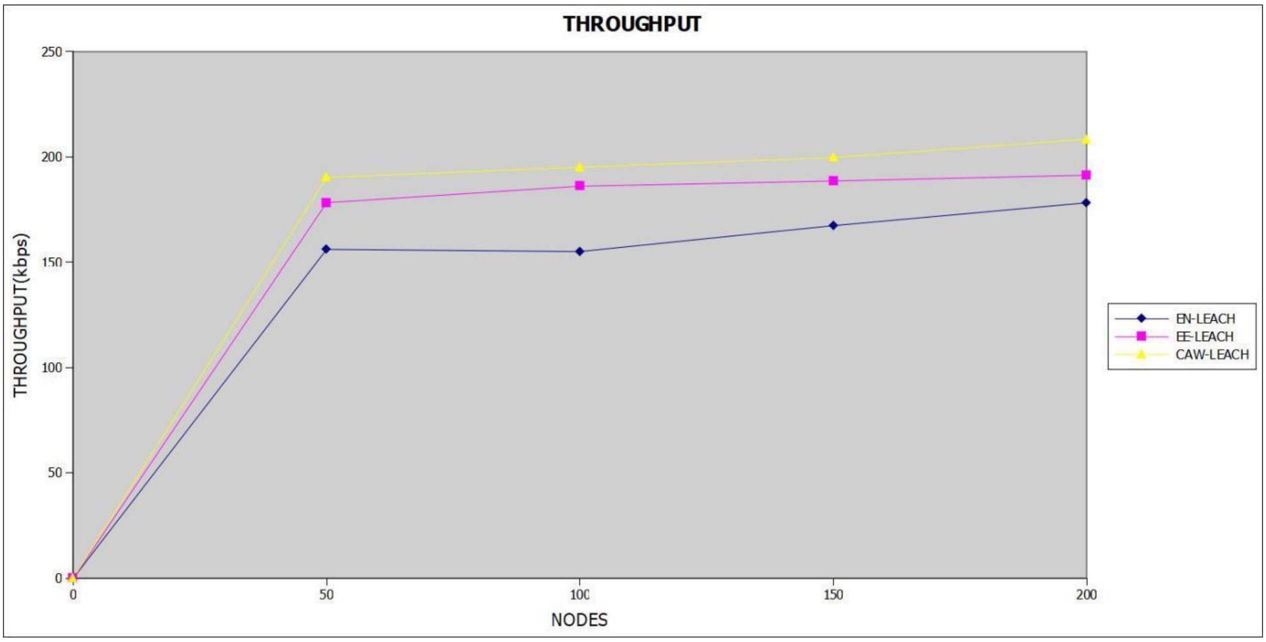

Fig 7: Throughput

Table 7: Throughput Comparison analysis of CAW-LEACH and existing methods as EN-LEACH and EE-LEACH

NODE	PROPOSED	EE-LEACH	EN-LEACH
50	190	178	156
100	195	186	155
150	199	188	167
200	208	191	178

The total number of data units that a node is able to handle in a specific amount of time is referred to as its throughput. The optimal selection of relays and energy-constrained CH selection through the use of DE both contribute to the best data aggregation. Table 7 shows that the proposed method has a higher throughput than existing methods. During the test, the suggested approach kept the average throughput rate as high as 389 kbps, whereas the existing methods kept a throughput rate that was lower than the proposed one. The graphical representation of throughput can be seen in Figure 7.

5. Conclusion

This work proposes an enhanced LEACH-based clustering approach to improve IoT based network data aggregation efficiency, conserving energy, and extending the network's lifetime. In the classic LEACH approach, the selection of cluster heads (CHs) is accomplished by the use of a random integer. Because LEACH accords the same priority to nodes with low residual energy as it does to nodes with high residual energy, this can lead to some nodes with low residual energy passing away before nodes with high residual energy. In addition, as IoT based networks are energy limited networks, congestion is the primary problem that needs to be addressed in order to achieve efficient data aggregation. We propose an improved LEACH protocol (CAW-LEACH) that takes into account the congestion indicator and estimated remaining energy of the nodes for CH selection and random number generation. This will help increase the energy efficiency as well as the stability of the CH. Its purpose is to ensure that the CH node that was just recently elected will not be given a second chance in this round. In addition, a congestion-aware data aggregation scheme is presented for the purpose of making data aggregation more effective. The simulation and analysis results show that CAW-LEACH can significantly outperform existing approaches for IoT-based networks in terms of energy efficiency, network lifetime, and data throughput.

References:

[1] Chen, J.I.Z. and Yeh, L.T., 2021. Graphene based Web Framework for Energy Efficient IoT

Applications. *Journal of Information Technology*, *3*(01), pp.18-28.

[2] Mamdiwar, S.D., Shakruwala, Z., Chadha, U., Srinivasan, K. and Chang, C.Y., 2021. Recent advances on IoT-assisted wearable sensor systems for healthcare monitoring. *Biosensors*, *11*(10), p.372.

[3] Maheshwari, P., Sharma, A.K. and Verma, K., 2021. Energy efficient cluster based routing protocol for WSN using butterfly optimization algorithm and ant colony optimization. *Ad Hoc Networks*, *110*, p.102317.

[4] Yuvaraj, N., Karthikeyan, T., Praghash, K. and Reddy, K.H., 2021. Binary flower pollination (BFP) approach to handle the dynamic networking conditions to deliver uninterrupted connectivity. *Wireless Personal Communications*, *121*(4), pp.3383-3402.

[5] Luo, J., Chen, Y., Wu, M. and Yang, Y., 2021. A survey of routing protocols for underwater wireless sensor networks. *IEEE Communications Surveys & Tutorials*, *23*(1), pp.137-160.

[6] Rani, K.S. and Balasubadra, K., 2022. Cross Layer Controlling Algorithm to Overcome Congestion in Underwater Wireless Sensor Network. *Wireless Personal Communications*, *122*(4), pp.3741-3756.

[7] Alfouzan, F.A., 2021. Energy-efficient collision avoidance MAC protocols for underwater sensor networks: Survey and challenges. *Journal of Marine Science and Engineering*, *9*(7), p.741.

[8] Priyadarshini, R.R. and Sivakumar, N., 2021. Cluster head selection based on minimum connected dominating set and bi-partite inspired methodology for energy conservation in WSNs. *Journal of King Saud University-Computer and Information Sciences*, *33*(9), pp.1132-1144.

[9] Jain, K., Mehra, P.S., Dwivedi, A.K. and Agarwal, A., 2022. SCADA: scalable cluster-based data aggregation technique for improving network lifetime of wireless sensor networks. *The Journal of Supercomputing*, pp.1-29.

[10] Roberts, M.K. and Ramasamy, P., 2022. Optimized hybrid routing protocol for energy-aware cluster head selection in wireless sensor networks. Digital Signal Processing, 130, p.103737.

[11] Singh, H. and Singh, D., 2021. Hierarchical clustering and routing protocol to ensure scalability and reliability in large-scale wireless sensor networks. The Journal of Supercomputing, 77(9), pp.10165-10183.

[12] Farahzadi, H.R., Langarizadeh, M., Mirhosseini, M. and FatemiAghda, S.A., 2021. An improved cluster formation process in wireless sensor network to decrease energy consumption. Wireless Networks, 27(2), pp.1077-1087.

[13] C. Gulzar, AmeenaYasmeen 2016. Maximum network lifetime with load balance and connectivity by clustering process for wireless sensor networks: International Journal of Computer Engineering In Research Trends ,3(7),pp.375-383.

[14] Bozorgi, S.M. and Bidgoli, A.M., 2019. HEEC: A hybrid unequal energy efficient clustering for wireless sensor networks. Wireless Networks, 25(8), pp.4751-4772.

[15] A Damayanthi ,Mohammad Riyaz Belgaum, 2022. A Study of Heterogeneity Characteristics over Wireless Sensor Networks . International Journal of Computer Engineering in Research Trends, 9(12), pp.258-262.

[16] Firdous, S., Bibi, N. and Wahid, M., 2021, December. An Energy-Efficient Cluster Based Routing Algorithm for Wireless Sensor Network. In 2021 International Conference on Frontiers of Information Technology (FIT) (pp. 182-187). IEEE.

[17] Khediri, S.E., Nasri, N., Khan, R.U. and Kachouri, A., 2021. An improved energy efficient clustering protocol for increasing the life time of wireless sensor networks. Wireless Personal Communications, 116(1), pp.539-558.

[18] Waz, I.R., Sobh, M.A. and Bahaa-Eldin, A.M., 2017, December. Internet of Things (IoT) security platforms. In 2017 12th International Conference on Computer Engineering and Systems (ICCES) (pp. 500-507). IEEE.

[19] Navdeep Kumar Chopra, Rajesh Kumar Singh., 2019, An energy-aware clustering approach for routing mechanism in WSN using Cuckoo Search. International Journal of Computer Engineering in Research Trends ,6(7),pp. 340-345.

[20] Mahapatra, C., Sheng, Z. and Leung, V.C., 2016, May. Energy-efficient and distributed data-aware clustering protocol for the Internet-of-Things. In 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-5). IEEE.

[21] Salem, A.O.A. and Shudifat, N., 2019. Enhanced LEACH protocol for increasing a lifetime of WSNs. Personal and Ubiquitous Computing, 23(5), pp.901-907.

[22] Dwivedi, A.K. and Sharma, A.K., 2021. EE-LEACH: Energy Enhancement in LEACH using Fuzzy Logic for Homogeneous WSN. Wireless Personal Communications, 120(4), pp.3035-3055.

A Review on Malware Analysis for IoT and Android System

Survey Article Published: 21 December 2022

Volume 4, article number 118, (2023) Cite this article

Chandra Shekhar Yadav & Sangeeta Gupta

Abstract

Today all humankind is willing to avail more facilities and hopes everything should be available with a click of the button. In order to offer different services, the developers have come with inbuilt modules of several systems. This make easy system develoment and services may be offered intantly. These services are connected to the internet and accessible via Android phones and IoT devices. But this inbuilt module suffers from a lot of vulnerabilities, bugs, and default settings which may be difficult to change, as happened at the time of changing the password of home-based Wi-Fi router, which require external applications and OTP verifications, etc. Due to these issues and new hacking tools and techniques, security is a major challenge today. The basic framework to provide adequate security of the system comprises five following principles: integrity, confidentiality, availability, privacy, and nonrepudiation. The attacker may leverage advantage of any shortcomings that may lead to several issues. This work explores the cause of threads/ vulnerability particularly for IoT, IIoT, SCADA, and Android application systems. The structure of this work is divided in different sections like, a short introduction to Malware, how it infects the system, and a detailed malware exploitation plan that is generally followed by expert attackers to exploit the vulnerabilities related to critical infrastructure or to defame the organization or countries is presented. In addition, General framework based introduction on IoT and Android is also presented with common vulnerabilities at every stage and respective mitigation strategies. Both static and dynamic analyses are evaluated in this work. It is identified that, for a better model design and evaluation, both are highly recommended for the implementation of effective malware detection strategies. Along with these models in order to protect the infra-structure Honeynet, IDS, IPS, Hardware-based securities like CPU and Memory and forensic analysis are also very effective.

Data availability

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

References

1. Kim Y, Kim I, Shim CY. A taxonomy for DOS attacks in VANET. In: 2014 14th International Symposium on Communications and Information Technologies (ISCIT), pp. 26–27. 2014.

2. Fontugne R, Borgnat P, Abry P, Fukuda K. Mawilab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. In: Proceedings of the 6th International Conference, pp. 1–12. 2010.

3. Ramilli M. Malware training sets: a machine learning dataset for everyone. 2016.

4. Kent AD. Cyber security data sources for dynamic network research. In: Dynamic Networks and Cyber-Security, World Scientific, pp. 37–65. 2016.

5. Ma J, Saul LK, Savage S, Voelker GM. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1245–1254. 2009.

6. Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In: 2015 military communications and information systems conference (MilCIS), pp. 1–6. 2015.

7. Turcotte MJM, Kent AD, Hash C. Unified host and network data set. In: Data Science for Cyber-Security, World Scientific, pp. 1–22. 2019.

8. Hagberg A, Lemons N, Kent A, Neil J. Connected components and credential hopping in authentication graphs. Tenth International Conf Signal Image Technol Based Syst. 2014;2014:416–23.

Google Scholar

9. Garcia S, Grill M, Stiborek J, Zunino A. An empirical comparison of botnet detection methods. Comput Secur. 2014;45:100–23.

Google Scholar

10. Parmisano A, Garcia S, Erquiaga M. Stratosphere laboratory. A labeled dataset with malicious and benign IoT network traffic. 2020.

11. Anderson HS, Roth P. Ember: an open dataset for training static pe malware machine learning models. 2018. arXiv:1804.04637.

12. Shaid SZM, Maarof MA. Malware behavior image for malware variant identification. Int Symp Biomet Secur Technol (ISBAST). 2014;2014:238–43.

Google Scholar

13. Weaver R. Visualizing and modeling the scanning behavior of the conficker botnet in the presence of user and network activity. IEEE Trans Inf Foren Secur. 2015;10(5):1039–51.

Google Scholar

14. Massicotte F, Couture M, Normandin H, Letourneau M. Navigating and visualizing the malware intelligence space. IEEE Netw. 2012;26(6):19–25.

Google Scholar

15. Kancherla K, Mukkamala S. Image visualization based malware detection. IEEE Symp Comput Intell Cyber Secur (CICS). 2013;2013:40–4.

Google Scholar

16. Koniaris I, Papadimitriou G, Nicopolitidis P, Obaidat M. Honeypots deployment for the analysis and visualization of malware activity and malicious connections. IEEE Int Conf Commun (ICC). 2014;2014:1819–24.

Google Scholar

17. Donahue J, Paturi A, Mukkamala S. Visualization techniques for efficient malware detection. IEEE Int Conf Intell Secur Inform. 2013;2013:289–91.

Google Scholar

18. Kirat D, Nataraj L, Vigna G, Manjunath BS. Sigmal: a static signal processing based malware triage. In: Proceedings of the 29th Annual Computer Security Applications Conference, pp. 89–98. 2013.

19. Bai L, Pang J, Zhang Y, Fu W, Zhu J. Detecting malicious behavior using critical api-calling graph matching. First Int Conf Inform Sci Eng. 2009;2009:1716–9.

Google Scholar

20. Blokhin K, Saxe J, Mentis D. Malware similarity identification using call graph based system call subsequence features. In: 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops, pp. 6–10. 2013.

21. Cesare S, Xiang Y, Zhou W. Control flow-based malware variantdetection. IEEE Trans Depend Secur Comput. 2013;11(4):307–17.

Google Scholar

22. Cesare S, Xiang Y. Malware variant detection using similarity search over sets of control flow graphs. In: 2011IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 181–189. 2011.

23. Atici MA, Sagiroglu S, Dogru IA. Android malware analysis approach based on control flow graphs and machine learning algorithms. In: 2016 4th International Symposium on Digital Forensic and Security (ISDFS), pp. 26–31. 2016.

24. Da Xu L, He W, Li S. Internet of things in industries: a survey. IEEE Trans Ind Inform. 2014;10(4):2233–43. https://doi.org/10.1109/TII.2014.2300753.

Article Google Scholar

25. Mosenia A, Jha NK. A comprehensive study of security of internet-of-things. IEEE Trans Emerg Top Comput. 2016;5(4):586–602.

Google Scholar

26. Yang Y, Wu L, Yin G, Li L, Zhao H. A survey on security and privacy issues in internet-of-things. IEEE Internet Things J. 2017;4(5):1250–8. https://doi.org/10.1109/JIOT.2017.2694844.

Article Google Scholar

27. Frustaci M, Pace P, Aloi G, Fortino G. Evaluating critical security issues of the IoT world: present and future challenges. IEEE Internet Things J. 2017;5(4):2483–95.

Google Scholar

28. Alaba FA, Othman M, Hashem IAT, Alotaibi F. Internet of things security: a survey. J Netw Comput Appl. 2017;88:10–28. https://doi.org/10.1016/j.jnca.2017.04.002.

Article Google Scholar

29. Khan MA, Salah K. IoT security: Review, blockchain solutions, and open challenges. Fut Gen Comput Syst. 2018;82:395–411. https://doi.org/10.1016/j.future.2017.11.022.

Article Google Scholar

30 Hameed S, Khan FI, Hameed B. Understanding security requirements and challenges in internet of things (IoT): A Review. J Comput Netw Commun. 2019. https://doi.org/10.1155/2019/9629381.

Article Google Scholar

31. Kouicem DE, Bouabdallah A, Lakhlef H. Internet of things security: a top–down survey. Comput Netw. 2018;141:199–221. https://doi.org/10.1016/j.comnet.2018.03.012.

Article Google Scholar

32. Riahi Sfar A, Natalizio E, Challal Y, Chtourou Z. A roadmap for security challenges in the Internet of Things. Digit Commun Netw. 2018;4(2):118–37. https://doi.org/10.1016/j.dcan.2017.04.003.

Article Google Scholar

33. Ziegeldorf JH, Morchon OG, Wehrle K. Privacy in the Internet of Things: threats and challenges. Secur Commun Netw. 2014;7(12):2728–42.

Google Scholar

34. Li C, Palanisamy B. Privacy in internet of things: from principles to technologies. IEEE Internet Things J. 2019;6(1):488–505. https://doi.org/10.1109/JIOT.2018.2864168.

Article Google Scholar

35. Boyes H, Hallaq B, Cunningham J, Watson T. The industrial internet of things (IIoT): an analysis framework. Comput Ind. 2018;101:1–12. https://doi.org/10.1016/j.compind.2018.04.015.

Article Google Scholar

36 Oztemel E, Gursev S. Literature review of Industry 4.0 and related technologies. J Intell Manuf. 2020;31(1):127–82. https://doi.org/10.1007/s10845-018-1433-8.

Article Google Scholar

37. Alcácer V, Cruz-Machado V. Scanning the Industry 4.0: a literature review on technologies for manufacturing systems. Eng Sci Technol Int J. 2019;22(3):899–919. https://doi.org/10.1016/j.jestch.2019.01.006.

Article Google Scholar

38. Fernández-Caramés TM, Fraga-Lamas P. A review on the use of blockchain for the Internet of Things. IEEE Access. 2018;6:32979–3001.

Google Scholar

39. Atlam HF, Alenezi A, Alassafi MO, Wills G. Blockchain with internet of things: Benefits, challenges, and future directions. Int J Intell Syst Appl. 2018;10(6):40–8.

Google Scholar

40. Reyna A, Martín C, Chen J, Soler E, Díaz M. On blockchain and its integration with IoT. Challenges and opportunities. Fut Gen Comput Syst. 2018;88:173–90. https://doi.org/10.1016/j.future.2018.05.046.

Article Google Scholar

41. Wang X, et al. Survey on blockchain for Internet of Things. Comput Commun. 2019;136:10–29. https://doi.org/10.1016/j.comcom.2019.01.006.

Article Google Scholar

42. Wang Q, Zhu X, Ni Y, Gu L, Zhu H. Blockchain for the IoT and industrial IoT: a review. Internet of Things. 2020;10: 100081. https://doi.org/10.1016/j.iot.2019.100081.

Article Google Scholar

43. Makhdoom I, Abolhasan M, Abbas H, Ni W. Blockchain's adoption in IoT: the challenges, and a way forward. J Netw Comput Appl. 2019;125:251–79. https://doi.org/10.1016/j.jnca.2018.10.019.

Article Google Scholar

44. Ferrag MA, Derdour M, Mukherjee M, Derhab A, Maglaras L, Janicke H. Blockchain technologies for the internet of things: research issues and challenges. IEEE Internet Things J. 2018;6(2):2188–204.

Google Scholar

45. Sengupta J, Ruj S, Das Bit S. A comprehensive survey on attacks, security issues and blockchain solutions for IoT and IIoT. J Netw Comput Appl. 2020. https://doi.org/

10.1016/j.jnca.2019.102481.

Article Google Scholar

46. Ahemd MM, Shah MA, Wahid A. IoT security: A layered approach for attacks and defenses. In: 2017 international conference on Communication Technologies (ComTech), pp. 104–110. 2017.

47. Andrea I, Chrysostomou C, Hadjichristofi G. Internet of Things: security vulnerabilities and challenges. IEEE Sympo Comput Commun (ISCC). 2015;2015:180–7.

Google Scholar

48. Ling Z, Liu K, Xu Y, Jin Y, Fu X. An end-to-end view of IoT security and privacy. In: 2017 IEEE Global Communications Conference, GLOBECOM 2017—Proceedings, Jul. 2017, vol. 2018–January, pp. 1–7. https://doi.org/10.1109/GLOCOM.2017.8254011.

49. Wurm J, Hoang K, Arias O, Sadeghi AR, Jin Y. Security analysis on consumer and industrial IoT devices. In: Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC, Mar. 2016, vol. 25–28-January-2016, pp. 519–524. https://doi.org/10.1109/ASPDAC.2016.7428064.

50. Zhang N, Mi X, Feng X, Wang X, Tian Y, Qian F. Understanding and mitigating the security risks of voice-controlled third-party skills on amazon alexa and google home. 2018. arXiv:1805.01525.

51. All IF. The 5 worst examples of Iot hacking and vulnerabilities in recorded history. 2017. https//www.iotforall.com/5-worst-iot-hacking-vulnerabilities.

52. Gomes T, Salgado F, Tavares A, Cabral J. Cute mote, a customizable and trustable end-

device for the internet of things. IEEE Sens J. 2017;17(20):6816–24.

Google Scholar

53 Porambage P, Schmitt C, Kumar P, Gurtov A, Ylianttila M. PAuthKey: a pervasive authentication protocol and key establishment scheme for wireless sensor networks in distributed IoT applications. Int J Distrib Sens Netw. 2014. https://doi.org/10.1155/2014/357430.

Article Google Scholar

54. Hei X, Du X, Wu J, Hu F. Defending resource depletion attacks on implantable medical devices. 2010. https://doi.org/10.1109/GLOCOM.2010.5685228.

55. Aman MN, Chua KC, Sikdar B. A Light-weight mutual authentication protocol for IoT systems. In: 2017 IEEE Global Communications Conference, GLOBECOM 2017—Proceedings, Jul. 2017, vol. 2018–January, pp. 1–6. https://doi.org/10.1109/GLOCOM.2017.8253991.

56. Choi J, Kim Y. An improved LEA block encryption algorithm to prevent side-channel attack in the IoT system. Asia-Pacific Signal Inform Process Assoc Annu Summit Confer (APSIPA). 2016;2016:1–4.

Google Scholar

57. Sicari S, Rizzardi A, Miorandi D, Coen-Porisini A. REATO: REActing TO denial of service attacks in the internet of things. Comput Netw. 2018;137:37–48. https://doi.org/10.1016/j.comnet.2018.03.020.

Article Google Scholar

58. Andrea I, Chrysostomou C, Hadjichristofi G. Internet of Things: Security vulnerabilities and challenges. In: Proceedings—IEEE Symposium on Computers and Communications, Feb. 2016, vol. 2016–February, pp. 180–187. https://doi.org/10.1109/ISCC.2015.7405513.

59. Varga P, Plosz S, Soos G, Hegedus C. Security threats and issues in automation IoT. 2017. https://doi.org/10.1109/WFCS.2017.7991968.

60. Guin U, Singh A, Alam M, Canedo J, Skjellum A. A secure low-cost edge device authentication scheme for the internet of things. In: Proceedings of the IEEE International Conference on VLSI Design, Mar. 2018, vol. 2018–January, pp. 85–90. https://doi.org/10.1109/VLSID.2018.42.

61. Glissa G, Rachedi A, Meddeb A. A secure routing protocol based on RPL for internet of things. 2016. https://doi.org/10.1109/GLOCOM.2016.7841543.

62. Pu C, Hajjar S. Mitigating Forwarding misbehaviors in RPL-based low power and lossy networks. In: CCNC 2018—2018 15th IEEE Annual Consumer Communications and Networking Conference, Mar. 2018, vol. 2018–January, pp. 1–6. https://doi.org/10.1109/CCNC.2018.8319164.

63. Cervantes C, Poplade D, Nogueira M, Santos A. Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things. In: Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015, Jun. 2015, pp. 606–611. https://doi.org/10.1109/INM.2015.7140344.

64. Shukla P. ML-IDS: a machine learning approach to detect wormhole attacks in Internet of Things. In: 2017 Intelligent Systems Conference, IntelliSys 2017, Mar. 2018, vol. 2018–January, pp. 234–240. https://doi.org/10.1109/IntelliSys.2017.8324298.

65. Airehrour D, Gutierrez JA, Ray SK. SecTrust-RPL: a secure trust-aware RPL routing protocol for Internet of Things. Fut Gen Comput Syst. 2019;93:860–76.

Google Scholar

66. Singh M, Rajan MA, Shivraj VL, Balamuralidhar P. Secure MQTT for Internet of Things (IoT). In: Proceedings—2015 5th International Conference on Communication Systems and Network Technologies, CSNT 2015, Sep. 2015, pp. 746–751. https://doi.org/10.1109/CSNT.2015.16.

67 Park N, Kang N. Mutual authentication scheme in secure internet of things technology for comfortable lifestyle. Sensors. 2016;16(1):20.

Google Scholar

68. Ashibani Y, Mahmoud QH. An efficient and secure scheme for smart home communication using identity-based signcryption. In: 2017 IEEE 36th International Performance Computing and Communications Conference, IPCCC 2017, Feb. 2018, vol. 2018–January, pp. 1–7. https://doi.org/10.1109/PCCC.2017.8280497.

69. Adat V, Gupta BB. A DDoS attack mitigation framework for internet of things. In: Proceedings of the 2017 IEEE International Conference on Communication and Signal Processing, ICCSP 2017, Feb. 2018, vol. 2018–January, pp. 2036–2041. https://doi.org/10.1109/ICCSP.2017.8286761.

70. Yin D, Zhang L, Yang K. A DDoS attack detection and mitigation with software-defined

internet of things framework. IEEE Access. 2018;6:24694–705. https://doi.org/10.1109/ACCESS.2018.2831284.

Article Google Scholar

71. Liu C, Cronin P, Yang C. A mutual auditing framework to protect IoT against hardware Trojans. In: Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC. 2016; 69–74. https://doi.org/10.1109/ASPDAC.2016.7427991.

72. Konigsmark STC, Chen D, Wong MDF. Information dispersion for trojan defense through high-level synthesis. In: Proceedings—2Design Automation Conference. 2016;05–09. https://doi.org/10.1145/2897937.2898034.

73. Naeem H, Guo B, Naeem MR. A light-weight malware static visual analysis for IoT infrastructure. In: 2018 International Conference on Artificial Intelligence and Big Data, ICAIBD 2018. 2018;240–244. https://doi.org/10.1109/ICAIBD.2018.8396202.

74. Su J, Danilo Vasconcellos V, Prasad S, Daniele S, Feng Y, Sakurai K. Lightweight classification of IoT malware based on image recognition. In: Proceedings—International Computer Software and Applications Conference. 2018;2:664–669. https://doi.org/10.1109/COMPSAC.2018.10315.

75. Chan M. Why cloud computing is the foundation of the Internet of Things. 2017.

76. Song T, Li R, Mei B, Yu J, Xing X, Cheng X. A privacy preserving communication protocol for IoT applications in smart homes. IEEE Internet Things J. 2017;4(6):1844–52.

Google Scholar

77. Machado C, Frohlich AA. IoT data integrity verification for cyber-physical systems using blockchain. In: Proceedings - 2018 IEEE 21st International Symposium on Real-Time Computing, ISORC 2018, pp. 83–90. 2018. https://doi.org/10.1109/ISORC.2018.00019.

78. Rahulamathavan Y, Phan RCW, Rajarajan M, Misra S, Kondoz A. Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption. In: 11th IEEE International Conference on Advanced Networks and Telecommunications Systems, ANTS 2017, pp. 1–6. 2018. https://doi.org/10.1109/ANTS.2017.8384164.

79. Zheng D, Wu A, Zhang Y, Zhao Q. Efficient and privacy-preserving medical data sharing in internet of things with limited computing power. IEEE Access. 2018;6:28019–27. https://doi.org/10.1109/ACCESS.2018.2840504.

Article Google Scholar

80. Gope P, Sikdar B. Lightweight and privacy-preserving two-factor authentication scheme for IoT devices. IEEE Internet Things J. 2018;6(1):580–9.

Google Scholar

81. Gai K, Choo KKR, Qiu M, Zhu L. Privacy-preserving content-oriented wireless communication in internet-of-things. IEEE Internet Things J. 2018;5(4):3059–67. https://doi.org/10.1109/JIOT.2018.2830340.

Article Google Scholar

82. Liu J, Zhang C, Fang Y. EPIC: a differential privacy framework to defend smart homes against internet traffic analysis. IEEE Internet Things J. 2018;5(2):1206–17. https://doi.org/10.1109/JIOT.2018.2799820.

Article Google Scholar

83. Esfahani A, et al. A lightweight authentication mechanism for M2M communications in industrial IoT environment. IEEE Internet Things J. 2019;6(1):288–96. https://doi.org/10.1109/JIOT.2017.2737630.

Article Google Scholar

84. Li X, Niu J, Bhuiyan MZA, Wu F, Karuppiah M, Kumari S. A robust ECC-Based provable secure authentication protocol with privacy preserving for industrial internet of things. IEEE Trans Ind Inform. 2018;14(8):3599–609. https://doi.org/10.1109/TII.2017.2773666.

Article Google Scholar

85. Srinivas J, Das AK, Wazid M, Kumar N. Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial Internet of Things. IEEE Trans Depend Secur Comput. 2018;17(6):1133–46.

Google Scholar

86. Yan Q, Huang W, Luo X, Gong Q, Yu FR. A multi-level DDoS mitigation framework for the industrial internet of things. IEEE Commun Mag. 2018;56(2):30–6. https://doi.org/10.1109/MCOM.2018.1700621.

Article Google Scholar

87. Sengupta S, Yasmin G, Ghosal A. Classification of male and female speech using

perceptual features. In: 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–7. 2017.

88. Sengupta J, Ruj S, Das Bit S. End to end secure anonymous communication for secure directed diffusion in IoT. In: ACM International Conference Proceeding Series, pp. 445–450. 2019. https://doi.org/10.1145/3288599.3295577.

89. Khanmohammadi K, Ebrahimi N, Hamou-Lhadj A, Khoury R. Empirical study of android repackaged applications. Empir Softw Eng. 2019;24(6):3587–629. https://doi.org/10.1007/s10664-019-09760-3.

Article Google Scholar

90. Google. "Google Play". 2020. https://play.google.com/store/apps?hl=fr.

91. Arp D, Spreitzenbarth M, Hubner M, Gascon H, Rieck K, Siemens C. Drebin: effective and explainable detection of android malware in your pocket. Ndss. 2014;14:23–6.

Google Scholar

92. Zhou Y, Jiang X. Dissecting android malware: characterization and evolution. In: Proceedings—IEEE Symposium on Security and Privacy, pp. 95–109, 2012. https://doi.org/10.1109/SP.2012.16.

93. Allix K, Bissyandé TF, Klein J, Le Traon Y. AndroZoo: collecting millions of Android apps for the research community. In: Proceedings—13th Working Conference on Mining Software Repositories, MSR 2016, pp. 468–471. 2016. https://doi.org/10.1145/2901739.2903508.

94. Parkour M. Contagio mobile. 2008.

95. VirusShare. VirusShare.com—because sharing is caring. 2011.

96. Suarez-Tangil G, Dash SK, Ahmadi M, Kinder J, Giacinto G, Cavallaro L. DroidSieve: fast and accurate classification of obfuscated android malware. In: CODASPY 2017—Proceedings of the 7th ACM Conference on Data and Application Security and Privacy, pp. 309–320. 2017. https://doi.org/10.1145/3029806.3029825.

97. Zangief. AppChina is the best Android app store alternative. 2014.

98. Maiorca D, Ariu D, Corona I, Aresu M, Giacinto G. Stealth attacks: an extended insight into the obfuscation effects on Android malware. Comput Secur. 2015;51:16–31. https://doi.org/10.1016/j.cose.2015.02.007.

Article Google Scholar

99. Zangief. Gfan provides you free Android apps and games. 2017. http://appcakefans.com/gfan-provides-you-free-android-apps-and-games/.

100. Chen T, Mao Q, Yang Y, Lv M, Zhu J. TinyDroid: A lightweight and efficient model for android malware detection and classification. Mob Inf Syst. 2018;2018. https://doi.org/10.1155/2018/4157156.

101. Chen J, Alalfi MH, Dean TR, Zou Y. Detecting android malware using clone detection. J Comput Sci Technol. 2015;30(5):942–56. https://doi.org/10.1007/s11390-015-1573-7.

Article Google Scholar

102. Potharaju R, Newell A, Nita-Rotaru C, Zhang X. Plagiarizing smartphone

applications: attack strategies and defense techniques. In: International symposium on engineering secure software and systems, pp 106–120. 2012.

103. Liu P, Wang W, Luo X, Wang H, Liu C. NSDroid: efficient multi-classification of android malware using neighborhood signature in local function call graphs. Int J Inf Secur. 2021;20(1):59–71. https://doi.org/10.1007/s10207-020-00489-5.

Article Google Scholar

104. Wang W, Gao Z, Zhao M, Li Y, Liu J, Zhang X. DroidEnsemble: detecting Android malicious applications with ensemble of string and structural static features. IEEE Access. 2018;6:31798–807.

Google Scholar

105. Zhou W, Zhou Y, Jiang X, Ning P. Detecting repackaged smartphone applications in third-party android marketplaces. 2012;317. https://doi.org/10.1145/2133601.2133640.

106. Qiao M, Sung AH, Liu Q. Merging permission and api features for android malware detection. In: Proceedings - 2016 5th IIAI International Congress on Advanced Applied Informatics, IIAI-AAI 2016. 2016; 566–571. https://doi.org/10.1109/IIAI-AAI.2016.237.

107. Wu DJ, Mao CH, Wei TE, Lee HM, Wu KP. DroidMat: android malware detection through manifest and API calls tracing. In: Proceedings of the 2012 7th Asia Joint Conference on Information Security, AsiaJCIS 2012. 2012;62–69. https://doi.org/10.1109/AsiaJCIS.2012.18.

108. Sarma B, Li N, Gates C, Potharaju R, Nita-Rotaru C, Molloy I. Android permissions: a perspective combining risks and benefits. In: Proceedings of ACM Symposium on

Access Control Models and Technologies, SACMAT. 2012;13–22. https://doi.org/10.1145/2295136.2295141.

109. Peng H, et al. Using probabilistic generative models for ranking risks of Android apps. In: Proceedings of the ACM Conference on Computer and Communications Security, pp. 241–252. 2012. https://doi.org/10.1145/2382196.2382224.

110. Enck W, Ongtang M, McDaniel P. On lightweight mobile phone application certification. In: Proceedings of the ACM Conference on Computer and Communications Security, pp. 235–245. 2009. https://doi.org/10.1145/1653662.1653691.

111. Aafer Y, Du W, Yin H. Droidapiminer: Mining api-level features for robust malware detection in android. In: International conference on security and privacy in communication systems, pp. 86–103. 2013.

112. Zhou Y, Wang Z, Zhou W, Jiang X. Hey, you, get off of my market: detecting malicious apps in official and alternative android markets. NDSS. 2012;25(4):50–2.

Google Scholar

113. Millar S, McLaughlin N, Martinez del Rincon J, Miller P. Multi-view deep learning for zero-day Android malware detection. J Inf Secur Appl. 2021;58. https://doi.org/10.1016/j.jisa.2020.102718.

114. Xiao X, Zhang S, Mercaldo F, Hu G, Sangaiah AK. Android malware detection based on system call sequences and LSTM. Multimed Tools Appl. 2019;78(4):3979–99. https://doi.org/10.1007/s11042-017-5104-0.

Article Google Scholar

115. Chaba S, Kumar R, Pant R, Dave M. Malware detection approach for android systems using system call logs. 2017.

116. Canfora G, Mercaldo F, Medvet E, Visaggio CA. Detecting android malware using sequences of system calls. In: 3rd International Workshop on Software Development Lifecycle for Mobile, DeMobile 2015—Proceedings. 2015;13–20. https://doi.org/10.1145/2804345.2804349.

117. Burguera I, Zurutuza U, Nadjm-Tehrani S. Crowdroid: behavior-based malware detection system for android. In Proceedings of the ACM Conference on Computer and Communications Security, pp. 15–25, 2011. https://doi.org/10.1145/2046614.2046619.

118. Feng P, Ma J, Sun C, Xu X, Ma Y. A novel dynamic android malware detection system with ensemble learning. IEEE Access. 2018;6:30996–1011. https://doi.org/10.1109/ACCESS.2018.2844349.

Article Google Scholar

119. Shabtai A, Kanonov U, Elovici Y, Glezer C, Weiss Y. 'Andromaly': a behavioral malware detection framework for android devices. J Intell Inf Syst. 2012;38(1):161–90. https://doi.org/10.1007/s10844-010-0148-x.

Article Google Scholar

120. Xie N, Zeng F, Qin X, Zhang Y, Zhou M, Lv C. RepassDroid: automatic detection of

android malware based on essential permissions and semantic features of sensitive APIs. In: Proceedings—2018 12th International Symposium on Theoretical Aspects of Software Engineering, TASE 2018, Dec. 2018, vol. 2018–January, pp. 52–59. https://doi.org/10.1109/TASE.2018.00015.

121. Wen L, Yu H. An Android malware detection system based on machine learning. AIP Conf Proceed. 2017;1864. https://doi.org/10.1063/1.4992953.

122. Bugiel S, Davi L, Dmitrienko A, Fischer T, Sadeghi A-R. Xmandroid: a new android evolution to mitigate privilege escalation attacks. Tech Univ Darmstadt Tech Rep TR-2011–04. 2011.

123. Bakour K, Ünver HM (2021) DeepVisDroid: android malware detection by hybridizing image-based features with deep learning techniques. Neural Comput Appl. 2021;1–18.

124. Falcone Y, Currea S, Jaber M (2013) Runtime verification and enforcement for android applications with RV-droid. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 7687 LNCS, pp. 88–95. 2013. https://doi.org/10.1007/978-3-642-35632-2_11.

125. Portokalidis G, Homburg P, Anagnostakis K, Bos H. Paranoid android: versatile protection for smartphones. In: Proceedings—Annual Computer Security Applications Conference, ACSAC, pp. 347–356. 2010. https://doi.org/10.1145/1920261.1920313.

126. Lee WY, Saxe J, Harang R. SeqDroid: obfuscated android malware detection using stacked convolutional and recurrent neural networks. In: Deep Learning Applications for Cyber Security, Springer, pp. 197–210. 2019.

127. Phu TN, Hoang LH, Toan NN, Tho ND, Binh NN. CFDVex: a novel feature extraction method for detecting cross-architecture IoT Malware. In: Proceedings of the Tenth International Symposium on Information and Communication Technology, pp. 248–254. 2019.

128. Islam R, Tian R, Batten LM, Versteeg S. Classification of malware based on integrated static and dynamic features. J Netw Comput Appl. 2013;36(2):646–56. https://doi.org/10.1016/j.jnca.2012.10.004.

Article Google Scholar

129. Shahzad F, Farooq M. ELF-Miner: using structural knowledge and data mining methods to detect new (Linux) malicious executables. Knowl Inf Syst. 2012;30(3):589–612. https://doi.org/10.1007/s10115-011-0393-5.

Article Google Scholar

130. Bai J, Yang Y, Mu S, Ma Y. Malware detection through mining symbol table of linux executables. Inf Technol J. 2013;12(2):380–4. https://doi.org/10.3923/itj.2013.380.384.

Article Google Scholar

131. HaddadPajouh H, Dehghantanha A, Khayami R, Choo KKR. A deep recurrent neural network based approach for internet of things malware threat hunting. Fut Gen Comput Syst. 2018;85:88–96. https://doi.org/10.1016/j.future.2018.03.007.

Article Google Scholar

132. Dovom EM, Azmoodeh A, Dehghantanha A, Newton DE, Parizi RM, Karimipour H. Fuzzy pattern tree for edge malware detection and categorization in IoT. J Syst Archit. 2019;97:1–7.

Google Scholar

133. Darabian H, Dehghantanha A, Hashemi S, Homayoun S, Choo KR. An opcode-based technique for polymorphic Internet of Things malware detection. Concurr Comput Pract Exp. 2020;32(6): e5173.

Google Scholar

134. Alhanahnah M, Lin Q, Yan Q, Zhang N, Chen Z. Efficient signature generation for classifying cross-architecture IoT malware. 2018. https://doi.org/10.1109/CNS.2018.8433203.

135. Alasmary H, Anwar A, Park J, Choi J, Nyang D, Mohaisen A. Graph-based comparison of IoT and android malware. In: International Conference on Computational Social Networks, pp. 259–272. 2018.

136. Alasmary H, et al. Analyzing and detecting emerging internet of things malware: a graph-based approach. IEEE Internet Things J. 2019;6(5):8977–88. https://doi.org/10.1109/JIOT.2019.2925929.

Article Google Scholar

137. Azmoodeh A, Dehghantanha A, Choo K-KR. Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning. IEEE Trans Sustain Comput. 2018;4(1):88–95.

Google Scholar

138. Nguyen H-T, Ngo Q-D, Le V-H. A novel graph-based approach for IoT botnet detection. Int J Inf Secur. 2020;19(5):567–77.

Google Scholar

139. Ngo QD, Nguyen HT, Le VH, Nguyen DH. A survey of IoT malware and detection methods based on static features. ICT Express, vol. 6, no. 4. Korean Institute of Communication Sciences, pp. 280–286. 2020. https://doi.org/10.1016/j.icte.2020.04.005.

140. Sikorski M, Honig A. Practical malware analysis: the hands-on guide to dissecting malicious software. No starch press. 2012.

141. Wang T-Y, Wu C-H. Detection of packed executables using support vector machines. In: 2011 International Conference on Machine Learning and Cybernetics. 2011;2:717–722.

142. Abimannan S, Kumaravelu R. A mathematical model of HMST model on malware static analysis. Int J Inf Secur Priv. 2019;13(2):86–103.

Google Scholar

143. Abdessadki I, Lazaar S. New classification based model for malicious PE files detection. Int J Comput Netw Inf Secur. 2019; 11(6).

144. Ligh MH, Adair S, Hartsteini B, Richard M. Malware analyst's cookbook and DVD. Wiley Publishing. 2011.

145. Liao X, Yuan K, Wang X, Li Z, Xing L, Beyah R. Acing the ioc game: toward automatic discovery and analysis of open-source cyber threat intelligence. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 755–766. 2016.

146. Schrittwieser S, Katzenbeisser S. Code obfuscation against static and dynamic reverse engineering. In: International workshop on information hiding, pp. 270–284. 2011.

147. Németh ZL. Modern binary attacks and defences in the windows environment—fighting against microsoft EMET in seven rounds. In: 2015 IEEE 13th International Symposium on Intelligent Systems and Informatics (SISY), pp. 275–280. 2015.

148. Cohen M. Scanning memory with Yara. Digit Investig. 2017;20:34–43.

Google Scholar

149. Sarantinos N, Benzaïd C, Arabiat O, Al-Nemrat A. Forensic malware analysis: the value of fuzzy hashing algorithms in identifying similarities. IEEE Trustcom/BigDataSE/ISPA. 2016;2016:1782–7.

Google Scholar

150. Gandotra E, Bansal D, Sofat S. Malware analysis and classification: a survey. J Inf Secur. 2014;2014.

151. Bidoki SM, Jalili S, Tajoddin A. PbMMD: a novel policy based multi-process malware detection. Eng Appl Artif Intell. 2017;60:57–70.

Google Scholar

152. Ndatinya V, Xiao Z, Manepalli VR, Meng K, Xiao Y. Network forensics analysis using Wireshark. Int J Secur Netw. 2015;10(2):91–106.

Google Scholar

153. Hoque N, Bhuyan MH, Baishya RC, Bhattacharyya DK, Kalita JK. Network attacks: taxonomy, tools and systems. J Netw Comput Appl. 2014;40:307–24.

Google Scholar

154. Eilam E. Reversing: secrets of reverse engineering. John Wiley & Sons. 2011.

155. Gibert Llauradó D, Mateu Piñol C, Planes Cid J. The rise of machine learning for detection and classification of malware: research developments, trends and challenge. J Netw Comput Appl. 2020;153:102526.

Google Scholar

156. Rathnayaka C, Jamdagni A. An efficient approach for advanced malware analysis using memory forensic technique. IEEE Trustcom/BigDataSE/ICESS. 2017;2017:1145–50.

Google Scholar

157. Kara I. A basic malware analysis method. Comput Fraud Secur. 2019;2019(6):11–9.

Google Scholar

158. Kävrestad J. Memory analysis tools. In: Fundamentals of Digital Forensics, Springer,

pp. 217–224. 2020.

159. Pirscoveanu RS, Hansen SS, Larsen TMT, Stevanovic M, Pedersen JM, Czech A. Analysis of malware behavior: type classification using machine learning. In: 2015 International conference on cyber situational awareness, data analytics and assessment (CyberSA), pp. 1–7. 2015.

160. Aslan Ö, Samet R. Investigation of possibilities to detect malware using existing tools. In: 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA), pp. 1277–1284. 2017.

161. Mirza QKA, Awan I, Younas M. CloudIntell: an intelligent malware detection system. Fut Gener Comput Syst. 2018;86:1042–53.

Google Scholar

162. Lin W, Lee D. Traceback attacks in cloud - Pebbletrace botnet. In: Proceedings - 32nd IEEE International Conference on Distributed Computing Systems Workshops, ICDCSW 2012, pp. 417–426. 2012. https://doi.org/10.1109/ICDCSW.2012.61.

163. Beuhring A, Salous K. Beyond blacklisting: cyberdefense in the era of advanced persistent threats. IEEE Secur Priv. 2014;12(5):90–3.

Google Scholar

164. Jiang J, Yasakethu L. Anomaly detection via one class svm for protection of scada systems. Int Conf Cyber Enabled Distrib Comput Knowl Discov. 2013;2013:82–8.

Google Scholar

165. Almalawi A, Yu X, Tari Z, Fahad A, Khalil I. An unsupervised anomaly-based detection approach for integrity attacks on SCADA systems. Comput Secur. 2014;46:94–110.

Google Scholar

166. O'Kane P, Sezer S, McLaughlin K, Im EG. SVM training phase reduction using dataset feature filtering for malware detection. IEEE Trans Inf Forens Secur. 2013;8(3):500–9.

Google Scholar

167. Torrisi NM, Vuković O, Dán G, Hagdahl S. Peekaboo: a gray hole attack on encrypted SCADA communication using traffic analysis. IEEE Int Confe Smart Grid Commu (SmartGridComm). 2014;2014:902–7.

Google Scholar

168. Nader P, Honeine P, Beauseroy P. ${l_p}$-norms in one-class classification for intrusion detection in SCADA systems. IEEE Trans Ind Inform. 2014;10(4):2308–17.

Google Scholar

169. Simmhan Y, et al. Cloud-based software platform for big data analytics in smart grids. Comput Sci Eng. 2013;15(4):38–47.

Google Scholar

170. Markel Z, Bilzor M. Building a machine learning classifier for malware detection. In: 2014 second workshop on anti-malware testing research (WATeR), pp. 1–4. 2014.

171. Nagano Y, Uda R. Static analysis with paragraph vector for malware detection. In:

Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication, pp. 1–7. 2017.

172. Huda S, et al. Defending unknown attacks on cyber-physical systems by semi-supervised approach and available unlabeled data. Inf Sci (Ny). 2017;379:211–28.

Google Scholar

173. Mohaisen A, Alrawi O, Mohaisen M. AMAL: high-fidelity, behavior-based automated malware analysis and classification. Comput Secur. 2015;52:251–66.

Google Scholar

174. Mira F, Brown A, Huang W. Novel malware detection methods by using LCS and LCSS. In: 2016 22nd International Conference on Automation and Computing (ICAC), pp. 554–559. 2016.

175. Shabtai A, Moskovitch R, Elovici Y, Glezer C. Detection of malicious code by applying machine learning classifiers on static features: a state-of-the-art survey. Inf Secur Tech Rep. 2009;14(1):16–29.

Google Scholar

176. Damodaran A, Di Troia F, Visaggio CA, Austin TH, Stamp M. A comparison of static, dynamic, and hybrid analysis for malware detection. J Comput Virol Hacking Tech. 2017;13(1):1–12.

Google Scholar

177. Barabosch T, Gerhards-Padilla E. Host-based code injection attacks: a popular technique used by malware. In: 2014 9th International Conference on Malicious and

Unwanted Software: The Americas (MALWARE), pp. 8–17. 2014.

178. Snow KZ, Rogowski R, Werner J, Koo H, Monrose F, Polychronakis M. Return to the zombie gadgets: undermining destructive code reads via code inference attacks. IEEE Symp Secur Priv (SP). 2016;2016:954–68.

Google Scholar

179. Lee B, Lu L, Wang T, Kim T, Lee W. From zygote to morula: fortifying weakened aslr on android. IEEE Symp Secur Priv. 2014;2014:424–39.

Google Scholar

180. Gisbert HM, Ripoll I. On the effectiveness of nx, ssp, renewssp, and aslr against stack buffer overflows. In: 2014 IEEE 13th International Symposium on Network Computing and Applications, pp. 145–152. 2014.

181. Rohlf C, Ivnitskiy Y. The security challenges of client-side just-in-time engines. IEEE Secur Priv. 2012;10(2):84–6.

Google Scholar

182. Van Der Veen V, et al. A tough call: mitigating advanced code-reuse attacks at the binary level. IEEE Symp Secur Priv (SP). 2016;2016:934–53.

Google Scholar

183. Xiao X, Yan R, Ye R, Li Q, Peng S, Jiang Y. Detection and prevention of code injection attacks on HTML5-based apps. Third Int Conf Adv Cloud Big Data. 2015;2015:254–61.

Google Scholar

184. Brookes S, Osterloh M, Denz R, Taylor S. The KPLT: the kernel as a shared object. In: MILCOM 2015–2015 IEEE Military Communications Conference, pp. 954–959. 2015.

185. Chen P, Wu R, Mao B. JITSafe: a framework against Just-in-time spraying attacks. IET Inf Secur. 2013;7(4):283–92.

Google Scholar

186. Kil C, Jun J, Bookholt C, Xu J, Ning P. Address space layout permutation (ASLP): towards fine-grained randomization of commodity software. In: 2006 22nd Annual Computer Security Applications Conference (ACSAC'06), pp. 339–348. 2006.

187. Hoekstra M, Lal R, Pappachan P, Phegade V, Del Cuvillo J. Using innovative instructions to create trustworthy software solutions. HASP@ ISCA. 2013;11(10): 2487726–2488370.

188. De la Hoz E, Cochrane G, Moreira-Lemus JM, Paez-Reyes R, Marsa-Maestre I, Alarcos B. Detecting and defeating advanced man-in-the-middle attacks against TLS. In: 2014 6th International Conference On Cyber Conflict (CyCon 2014), pp. 209–221. 2014.

189. Buhov D, Huber M, Merzdovnik G, Weippl E. Pin it! Improving Android network security at runtime. In: 2016 IFIP Networking Conference (IFIP Networking) and Workshops, 2016, pp. 297–305.

190. Merzdovnik G, Buhov D, Voyiatzis AG, Weippl ER (2016) Notary-assisted certificate pinning for improved security of android apps. In: 2016 11th International Conference on Availability, Reliability and Security (ARES), pp. 365–371. 2016.

191. Jiang S, Li W, Li H, Zhang Y, Zhang H, Liu Y. Fault localization for null pointer exception based on stack trace and program slicing. In: 2012 12th International Conference on Quality Software, pp. 9–12. 2012.

192. Romano D, Di Penta M, Antoniol G. An approach for search based testing of null pointer exceptions. In: 2011 Fourth IEEE International Conference on Software Testing, Verification and Validation, pp. 160–169. 2011.

193. Ma S, Jiao M, Zhang S, Zhao W, Wang DW. Practical null pointer dereference detection via value-dependence analysis. IEEE Int Symp Softw Reliab Eng Worksh (ISSREW). 2015;2015:70–7.

Google Scholar

194. Hsu F-H, Tso C-K, Yeh Y-C, Wang W-J, Chen L-H. Browserguard: a behavior-based solution to drive-by-download attacks. IEEE J Sel areas Commun. 2011;29(7):1461–8.

Google Scholar

195. Cheng H, Yong F, Liang L, Wang L-R. A static detection model of malicious PDF documents based on naive Bayesian classifier technology. Int Conf Wavelet Act Media Technol Inform Proces (ICWAMTIP). 2012;2012:29–32.

Google Scholar

196. Al-Taharwa IA, Lee H-M, Jeng AB, Ho C-S, Wu K-P, Chen S-M. Drive-by disclosure: a large-scale detector of drive-by downloads based on latent behavior prediction. In: 2015 IEEE Trustcom/BigDataSE/ISPA, vol. 1, pp. 334–343. 2015.

197. Welch I, Gao X, Komisarczuk P. Detecting heap-spray attacks in drive-by downloads: Giving attackers a hand. In: 38th Annual IEEE Conference on Local Computer

Networks, pp. 300–303. 2013.

198. Malipatlolla S, Feller T, Shoufan A, Arul T, Huss SA. A novel architecture for a secure update of cryptographic engines on trusted platform module. Int Conf Field-Program Technol. 2011;2011:1–6.

Google Scholar

199. Maybaum M, Toelle J. ARMing the trusted platform module pro-active system integrity monitoring focussing on peer system notification. In: MILCOM 2015–2015 IEEE Military Communications Conference, pp. 1584–1589. 2015.

200. Razmi MAY, Hashim H. Forming virtualized test bed for Trusted Platform Module in Windows environment. IEEE Int Conf Comput Appl Ind Electron (ICCAIE). 2011;2011:645–50.

Google Scholar

201. Yu Z, Wang Q, Zhang W, Dai H. A cloud certificate authority architecture for virtual machines with trusted platform module. In: 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, pp. 1377–1380. 2015.

202. Kim D, Jeon Y, Kim J. A method based on platform integrity verification for activating a mobile trusted module. Int Conf Inform Commun Technol Converg (ICTC). 2015;2015:1174–6.

Google Scholar

203. Zhang F, Leach K, Sun K, Stavrou A. Spectre: A dependable introspection framework

via system management mode. In: 2013 43rd Annual IEEE/IFIP international conference on dependable systems and networks (DSN), pp. 1–12. 2013.

204. Messaoud BID, Guennoun K, Wahbi M, Sadik M. Advanced persistent threat: new analysis driven by life cycle phases and their challenges. Int Conf Adv Commun Syst Inform Secur (ACOSIS). 2016;2016:1–6.

Google Scholar

205. Mahboob A, Zubairi JA. "Securing SCADA systems with open source software. High Cap Opt Netw Emerg Enabl Technol. 2013;2013:193–8.

Google Scholar

206. Jain P, Tripathi P. SCADA security: a review and enhancement for DNP3 based systems. CSI Trans ICT. 2013;1(4):301–8.

Google Scholar

207. Yang Y, McLaughlin K, Littler T, Sezer S, Wang HF. Rule-based intrusion detection system for SCADA networks. 2013.

208. MacDermott Á, Shi Q, Merabti M, Kifayat K. Intrusion detection for critical infrastructure protection. 2012.

209. Yang Y, McLaughlin K, Sezer S, Yuan YB, Huang W. Stateful intrusion detection for IEC 60870-5-104 SCADA security. In: 2014 IEEE PES General Meeting| Conference & Exposition, pp. 1–5. 2014.

210. Oman P, Phillips M. Intrusion detection and event monitoring in SCADA networks.

In: International Conference on Critical Infrastructure Protection, pp. 161–173. 2007.

211. Fovino IN, Coletta A, Carcano A, Masera M. Critical state-based filtering system for securing SCADA network protocols. IEEE Trans Ind Electron. 2011;59(10):3943–50.

Google Scholar

212. Kirsch J, Goose S, Amir Y, Wei D, Skare P. Survivable SCADA via intrusion-tolerant replication. IEEE Trans Smart Grid. 2013;5(1):60–70.

Google Scholar

213. Carcano A, Coletta A, Guglielmi M, Masera M, Fovino IN, Trombetta A. A multidimensional critical state analysis for detecting intrusions in SCADA systems. IEEE Trans Ind Inform. 2011;7(2):179–86.

Google Scholar

214. Winn M, Rice M, Dunlap S, Lopez J, Mullins B. Constructing cost-effective and targetable industrial control system honeypots for production networks. Int J Crit Infrastruct Prot. 2015;10:47–58.

Google Scholar

215. Baecher P, Koetter M, Holz T, Dornseif M, Freiling F. The nepenthes platform: an efficient approach to collect malware. In: International Workshop on Recent Advances in Intrusion Detection, pp. 165–184. 2006.

216. Disso JP, Jones K, Bailey S. A plausible solution to SCADA security honeypot systems. In: 2013 Eighth International Conference on Broadband and Wireless Computing, Communication and Applications, pp. 443–448. 2013.

217. Pham V-H, Dacier M. Honeypot trace forensics: the observation viewpoint matters. Futur Gener Comput Syst. 2011;27(5):539–46.

Google Scholar

218. Brand M, Valli C, Woodward A. A threat to cyber resilience: a malware rebirthing botnet. 2011.

219. Goldenberg J, Shavitt Y, Shir E, Solomon S. Distributive immunization of networks against viruses using the 'honey-pot'architecture. Nat Phys. 2005;1(3):184–8.

Google Scholar

220. Erol-Kantarci M, Mouftah HT. Smart grid forensic science: applications, challenges, and open issues. IEEE Commun Mag. 2013;51(1):68–74.

Google Scholar

221. Amnesty International Security Lab. Mobile Verification Toolkit. 2021. https://docs.mvt.re/en/latest/ (Accessed Aug. 12, 2021).

Author information

Authors and Affiliations

STQC, Ministry of Electronics and Information Technology, Delhi, India

Chandra Shekhar Yadav

Department of Computer Science and Engineering, Chaitanya Bharathi Institute of Technology, Hyderabad, India
Sangeeta Gupta

Corresponding author

Correspondence to Chandra Shekhar Yadav.

Ethics declarations

Conflict of Interest

Author declares that they has no conflict of interest.

Ethical Approval

This article does not contain any studies with human participants performed by any of the authors.

Additional information

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Supplementary Information

Below is the link to the electronic supplementary material.

Supplementary file1 (DOCX 696 kb)

Rights and permissions

About this article

Cite this article

Yadav, C.S., Gupta, S. A Review on Malware Analysis for IoT and Android System. *SN COMPUT. SCI.* **4**, 118 (2023). https://doi.org/10.1007/s42979-022-01543-w

Received	Accepted	Published
31 December 2021	10 November 2022	21 December 2022

DOI

https://doi.org/10.1007/s42979-022-01543-w

Keywords

IoT IIoT Android Malware Virus SCADA IDS IPS