Hybrid Machine Learning model in efficient crowd detection system for MCS application

Pinky Dagar[1], Dr. Mukesh Kumar2

[1]Research Scholar, Department of CSE, NIILM University, Kaithal
[2]Professor, Department of CSE, NIILM University, Kaithal

Abstract: **Portable crowd sensing (MCS) system called EMC3, which means to decrease energy utilization of individual client just as all participants in information move brought about by task and information assortment of MCS undertakings, considering the client security issue, insignificant number of errand task necessity and sensing region inclusion imperative. Specifically, EMC incorporates novel speed control and dynamic instruments for task, utilizing participants' present call, verifiable call records just as anticipated future calls and versatility, to guarantee the normal number of participants to return detected outcomes and completely cover the objective region, with the target of allocating a negligible number of assignments.. In this paper, we proposed Hybrid Machine Learning model in efficient crowd detection system for MCS application. It shows more accuracy than existing system.**

Keywords: **Smartphones, Mobile Crowd Sensing, Wireless Sensor Networks, applications Internet of Things, Bluetooth,**

I. INTRODUCTION

To provide representative models, Creekwatch enables observation of watersheds through crowd detection data, such as measurement of stream bed water, bank debris, and river flow. The Singapore National Environment Agency uses Haze Watch to take advantage of the swarm link for aerial observation. Executive waste is a major problem in the smart, manageable urban spaces of the future. Trash Watch and Waste App are two apps that residents use to filter waste from bins to further develop reuse. Accelerometers connected to mobile phones above moving vehicles can help detect vibrations from the MCS range, allowing on-board traffic to be distinguished and free parking spaces to be identified, as states, thanks to sensors.

Energy efficiency in MCS systems

Providing answers for more sustainable and environmentally friendly improvements is the most difficult problem we face around the world. Under these unique circumstances, the implementation of effective and sophisticated ICT frameworks can significantly reduce energy consumption. The Greener Future Degree appeals to organizations, individuals, public foundations, and governments for a variety of reasons. First of all, energy production reliably affects the climate and requires a reduction in carbon footprint and gas emissions. Second, and important for businesses in general, the practical breakthrough has a financial impact on bills. Also, organizations engage with green readers to brag about their available and more attractive image to customers. Today, scientists are making an extraordinary attempt to explore power generation methods in broadcast recording systems and correspondence organizations. In particular, ideal data classification models, which modestly influence power consumption, represent an extraordinary research focus.

In this particular situation, MCS Crusades requires a lot of data from shiny devices and battery consumption should be as low as possible so as not to prevent members from providing data. Most of the energy consumed by mobile devices is dedicated to recognizing and announcing activity based on selected sensors and matching contacts. The energy used for transportation is usually more critical than that of transportation. In MCS frameworks, power efficiency can be viewed as a trade-off between the device battery channel and key mission performance indicators such as data consistency, space confinement, data metering. The MCS worldview also helps foster persistent data classification systems, which you should stay away from. Waste of batteries and increase in the contribution of households.

Distributed computing systems for MCS services

In MCS campaigns, sharp gear provides a lot of data to worry about, but neighborhood building provides irrelevant skills. Transmission of data to broadcast logging facilities for processing and investigation is a generally favorable game plan that maintains information from the swarm. Appropriate registry systems include ideal models that break down computational problems into small tasks performed by different substances with the aim of further promoting performance and capabilities. Today, MCS applications regularly take advantage of circular processing, multiple access edge configuration (MEC), and haze are among some suitable systems. Distributed logging involves passing a shared compute and storage structure, securing payload data with a ubiquitous array technique.

Be that as it may, the widespread proliferation of edged devices makes it difficult for the prospect of excellent processing to meet the ever-increasing world-class requirements for low-downtime mobile applications. To do this, it is necessary to understand the degree of fog and the ideal MEC patterns. Closer to end customers, which is a generally interesting technique for data blending policies like MCS that need to execute exercises quickly. Cisco offered Haze Enrollment as a cloud development. Estimation and comparison (for example, base stations, routes and inputs).
Its peculiarity is that some layers are close to the end customer. The European Telecommunications Standards Institute (ETSI) standardized the MEC in mid-2017, changing the way we view the mobile edge with multi-access edge processing to include it as an enabling impact of different access paths. The bounce distance between the end user and the destination of the Mobile Manager mappings To show several representative patterns, EdgeSense is an MEC-based MCS framework that leverages a broadcast mapping for periodic notifications. RMCS is a robust MCS scheme that consolidates MEC resources and important learning content to limit transmission lethargy.

II. EXISTING WORKS

"The consumer of the product can collect data in a concise manner either by himself or with the help of the customer. In some cases, the main mobile client may collect data by interacting with additional IoT clients, which are often not suitable for directly interacting with the application worker. The MCS design is technically but comprehensively illustrated in Figure 1.3. Look at the work of Louta et al. [2012] for further discussion on joint MCS projects. Figure 1.4 shows a more detailed representation of each possible phase of the CrowdSensing interaction, which is more accurate than the previous representation. There have been several attempts to develop a suitable reference measurement design to delineate useful principal squares and explain the main processes of MCS frameworks, but none have been successful. The work of these individuals forms the basis of the following investigation. Less

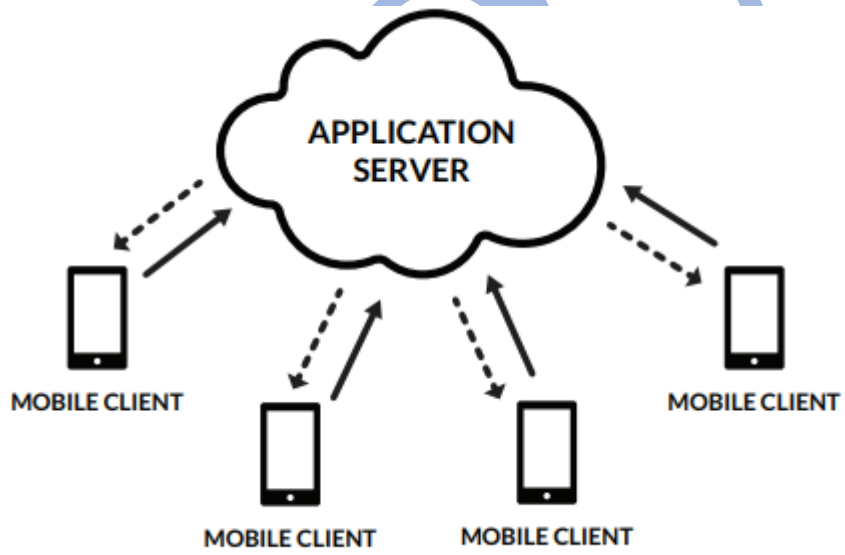

Figure 1 MCS application software architecture.

Anal Promises has been removed to make the published cycle more current and reasonable in light of some new work discussed so far.
Start of activity At this stage, the MCS contact point or coordinator assigns the defined responsibilities and provides a detailed description of the actions to be carried out. Customers can also start the project development process; These are the same people who consume the information obtained through the MCS application. It is possible that the representation is in a common language or it is an explicit language in the space that the consumers of the product can understand and give to the volunteers depending on the scene used. In some cases, the creation of the competition is verified in the structure of practices Volunteer researchers who sign up for the app are tasked with performing an identified screening activity.

Distribution of tasks in a project Depending on the circumstances, this phase can be carried out in a centralized or decentralized manner. It is possible that the central element subdivides and delegates the task of perception to specific members (or centers of organization of perception), while it may try to respect certain limits, for example of race, followed by the number of volunteers required under a certain margin, which guarantee a negligible normal trust between the selected members, etc. The central element can delegate the recognition task to explicit members (or recognition centers). Another approach is to inform all customers that another trip is available and give them the opportunity to decide whether or not to continue.

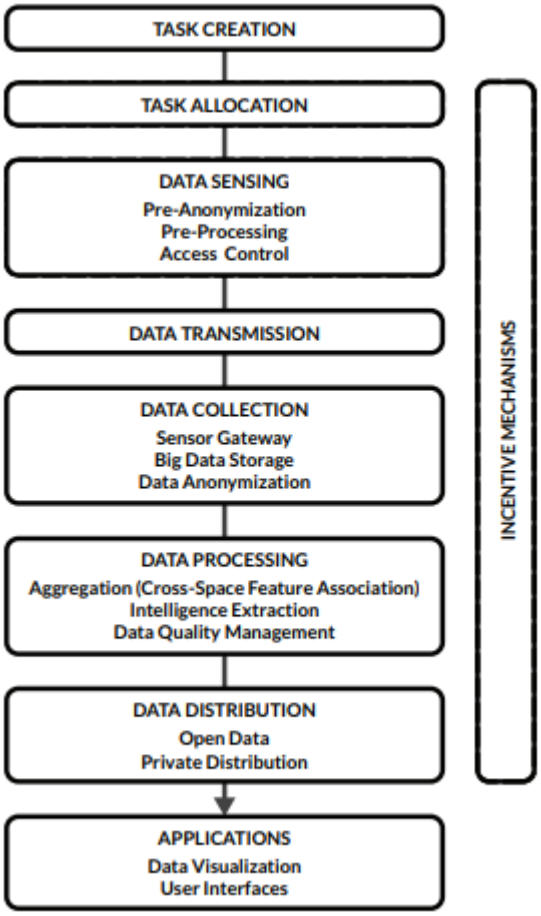

Figure 21Proposed structure of the reference process for MCS applications.

Participate in the perception task or do without Depending on the structure used in the organization to inspire the motivational forces, several options can be allowed, such as B. Activities based on hurdles. In terms of data collection, this includes both data collected from mobile devices and data provided by customers through mobile Internet applications. By their very nature, MCS applications face security and safety concerns. Therefore, it is important to provide customers with automated or semi-automated tools to determine what type of information to share and with whom to distribute it. Many MCS frameworks rely on access control components and pre-anonymization techniques to achieve their goals. To reduce transmission costs and reduce data size, data is typically pre-processed on the customer's own device. A major hurdle for an aMCS application is determining the appropriate tradeoff between the processing effort that must be performed locally on mobile phones and in the cloud after data delivery.

III. PROPOSED SYSTEM ARCHITECTURE

Our winning concept was that when you use machine learning, you start with the data and let the data guide you through the reasoning process to reach a conclusion. What strategies does a company use to achieve its goals? More importantly, since everything is running and shipping demanding applications, an orchestration cycle is required to identify the business problem to be solved and acquire the appropriate data sources. How does this method of managing application development affect the business? When developing logic-driven applications, it is recognized that business cycles will continue to run in the background. Either way, it's a big change. If you can start presenting facts, you can make changes to your cycle and reasoning. Machine learning can make the job design more remarkable and believable by using data mining techniques if necessary.

ROLE OF ALGORITHMS

It would be impossible to talk about machine learning without including a part in the estimates. There are many principles in computing that tell a computer how to communicate, control, and manipulate data in the most efficient way possible. When it comes to calculations, they can be as simple as using a strategy to add a number segment or as complex as seeing a person's face in a photograph. For a calculation to make sense, it must be written in the form of a program that computers can understand. Machine learning computations are regularly written in one of three programming languages: Java, Python, or R. These languages are used to write machine learning algorithms. All of these languages are a collection of machine learning libraries that help compute a collection of machine learning algorithms. It should also be noted that these languages have vibrant customer networks that are constantly providing code and discussing considerations, pitfalls, and ways to solve business problems. There are some differences between machine learning estimation and other types of

calculations. For most cost estimates, a designer starts with the calculation. Either way, machine learning is overturning the traditional collaborative model. In machine learning, the model is built using data from the real world. Increase the amount of data included in the estimate and the better the result. You can continuously perform a defined calculation by exposing the machine learning estimator to a gradually increasing set of data points.

TYPES OF MACHINE LEARNING ALGORITHMS

Choosing the most appropriate calculation is a combination of science and manual work. For example, two data analysts tasked with the same business problem may generate two very different estimates of how to go in the same direction as each other. Either way, familiarity with the many types of machine learning predictions can help data analysts identify the most ideal type of calculation to perform. This section provides a high-level overview of the main types of machine learning predictions.

Bayesian

With Bayesian estimation, data scientists are free to incorporate prior beliefs about what non-data-driven models should look like. It is reasonable to wonder why people are so enthusiastic about Bayesian calculations when so much emphasis is placed on the data that describes the model. These calculations are especially important when you don't have access to large amounts of data that can be used to build a complete model. For example, in the case of a Bayesian computation, it would be helpful if you had data that preceded part of the model and could write it unambiguously. And if we were to use the framework of a clinical imaging evaluation system that looks for lung problems, if a popular magazine article evaluates the probability of certain lung problems based on lifestyle, those probabilities could be coded into the model.

Grouping

A very simple approach to collection is grouping, which consists of grouping elements near their edges (in a group). All items in a bundle are more similar to each other than items in other groups. In this context, without a name in the dataset, bundling is considered a kind of self-directed learning. The estimate includes all-inclusive limits and thus wraps them appropriately.

Decision Tree

Choice tree calculations describe the final results of a decision by using a development plan to represent the possible outcomes of a decision. You can use decision trees to plan the expected consequences of a decision. An expected outcome is the center of each node in the decision tree. Rates are assigned to certain key points based on the probability of the outcome occurring.

Decision trees are sometimes used to guide business initiatives. You may need to assess the impact of a 20% discount on customers and the potential results. Customers can be divided into four categories:

I. Convincing who is likely to buy from you if you get a job
II. Safe things you buy anyway
III. Lost causes that will never buy
IV. Fragile clients who may have conflicting responses to a process

If you give up an exposure effort, you clearly need to make an effort not to submit articles to three of the socials, as they may not respond at all or very well. When you focus on what's interesting, you get the most out of your business (ROI). A decision tree helps you map these four customer social networks and determine prospects and customers based on who will respond best to the current effort.

REDUCTION OF DIMENSIONS

Dimensionality reduction helps systems get rid of data that is no longer relevant to further investigation. These evaluations are done in a social setting and are used to screen out repetitive data, quirks, and other irrelevant data. It can be helpful to reduce the dimensionality of your data as you explore sensor data and other Internet of Things (IoT) use cases. In Internet of Things systems, there can be a large number of data hubs that simply indicate that a sensor has been triggered. There is no need to look after and review this "on" data and it will consume a lot of additional disk space. Also, removing this stale data would greatly improve the presentation of a machine learning system. Finally, reducing the dimensionality of the data helps analysts in their ability to visualize information.

Instance-based

Case-based estimation is used when you need to classify new data guides that may have similarities in the data production process. Since there is no preliminary stage in this estimation strategy, it is commonly known as the torpid ship plan. Even after everything is considered, model-based estimation essentially relates the new data to the training data and arranges the coordinates of the new data in a way that is consistent with similarity to the training data. Case-based learning is not appropriate for data sets that contain a non-homogeneous combination of variables, data that is considered useless, or data that lacks functionality. When it comes to applying a plan, case calculations can be very helpful. Pattern learning is used in a variety of applications, such as B. Composite and regular design evaluation and spatial research. In the disciplines of life sciences, pharmacy, science and planning, estimates based on individual models are often used for evaluation.

Train machine learning systems

You can start with an iterative process of building and refining a model, choosing the most appropriate estimator, building and testing a structure. It is essential for the development of machine learning measures that are well prepared. More specifically, if you are building a machine learning system, you know the interactions (eg, customer payment habits and purchase history, geography, etc.), and you know your ideal goal (predicting a customer's propensity to mix). It doesn't matter what you think about the data; What is amazing is the mathematical ability to turn it into a prediction of the customer's heartbeat. After a certain amount of customer data has been sent to the learning estimator, the system becomes more accurate in predicting the likelihood of a customer mix. The process of preparing a machine learning calculation to create an accurate model can be divided into three stages:

Representation.

The calculation creates a model that can be used to convert the provided data into the best possible results. As the learning algorithm is exposed to additional data, it gains capacity through the connection between the raw data and the data

centers, which are the most reliable predictors of the best outcome, as shown in Figure 1.

Evaluation

To score and score the different models produced by the estimator, a person or computer must score and score the models based on which model provides the most reliable predictions. It is important to remember that once the model is functionally tested, it will receive obfuscation data. Make sure the model is compressed and does not exceed your build data.

Optimization

After the estimation process has built and evaluated many models, choose the estimate that works best. If you expose your calculation to a wider variety of data levels, choose the most concise model.

The most important part of the training cycle is having enough data to be able to test the model and see how it performs. Initial attempts at preparation often give mixed results. This means that you may need to modify the model or provide additional data to be successful.

V. RESULTS

"The popularity of applications and the study of usage patterns are important elements in understanding how mobile phones are used. Several studies of flexible app usage have identified factors that influence download decisions but none have attempted to determine what happens after the app slows down. In the past, several researchers have focused on common usage and how it is affected by contextual features. Several survey organizations use maintenance fees1 to demonstrate the success of their applications. We believe that our assessment in Manuscript I in the appendix is the most comprehensive and independently describes in detail what happens after the application is submitted".

Retention rate

The number of customers who continue to use the application d days after the first use is called the degree of consistency on day d. To assess the severity of a customer and question, we record the first and last submission of the question by the customer for each customer and for each question.

Figure 5.1 illustrates the introduction of different levels of consistency during the main week. First-day reliability benchmarks for apps with fewer than 10 users are nearly half of those for apps with more than 10 users, versus 80% reported by various testing organizations2. The confidence threshold is 62% for applications with a thousand or more users. First-day consistency reaches 68% for the top 100 apps, and the seven-day reliability standard for the top 100 apps is more than half that. This analysis shows that consistency criteria often depend on the underlying number of customers and that well-known programs maintain customer integrity over long periods of business.

Trend analysis.

While retention rate is a measure of an app's long-term attractiveness to individual users, it doesn't take into account the app's immediate popularity, usage trends, or seasonal patterns. Figure 5.2 illustrates the usage patterns of various programs that were automatically selected (according to the peak detection algorithm) from the first day of use to the 100th day of use, as shown in the graph. The example can be used to demonstrate that application usage patterns do not necessarily follow a simple release pattern as predicted by retention rates. Figure 5.2 shows a series of uptrends followed by a series of downtrends. These apps have regained their relevance for a variety of reasons, some of which are unknown.

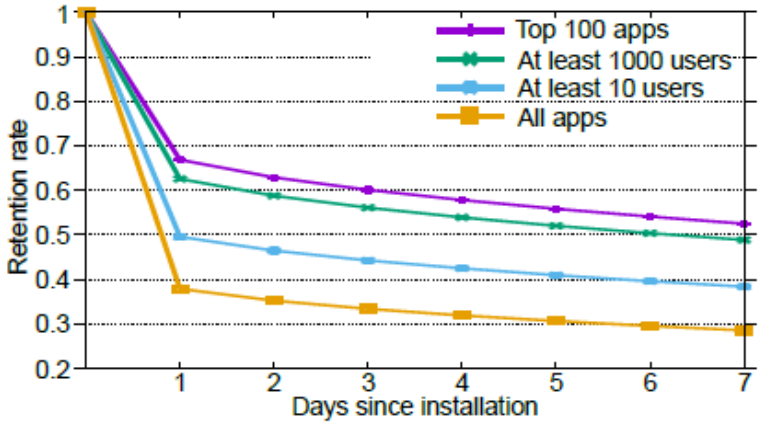

Figure 3 1Retention rate in the first week

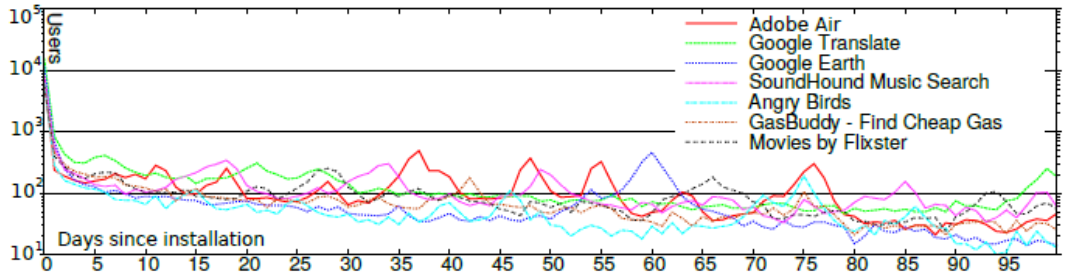

Figure 4 2Application usage patterns up to 100 days

VI. CONCLUSION & FUTURE WORK

Based on these facts, we build in [61] a technique to determine the software life cycle. Flop, Hot, Dominant and Marginal trades are the categories we use to classify trending trading patterns in our research. Figure 5.3 shows three types of crosses: flop, hot, and dominant, with examples of each. When we apply this trend analysis to the Carat dataset, we find that 40% of apps are edge apps, which have a relatively small overall user base. For the remaining 60% of the population, the following patterns can be identified: 0.4% are dominant, meaning they consistently achieve high popularity, 1% are failures, meaning they constantly lose popularity, and 7% is attractive, this is his increasingly popular name. As we will see in more detail below, this information can be used for application recommendation systems.

Reference

[1] Bin Guo, Zhiwen Yu, Xingshe Zhou, and Daqing Zhang. From par- ticipatory sensing to mobile crowd sensing. In Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on, pages 593–598. IEEE, 2014.

[2] Dejun Yang, GuoliangXue, Xi Fang, and Jian Tang. Incentive mecha- nisms for crowdsensing: Crowdsourcing with smartphones. Biological Cybernetics, 24(3):1732–1744, 2016.

[3] Raghu K Ganti, Fan Ye, and Hui Lei. Mobile crowdsensing: current state and future challenges. IEEE Communications Magazine, 49(11), 2011.

[4] HV Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Pa- pakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. Big data and its technical challenges. Communications of the ACM, 57(7):86–94, 2014.

[5] Daniel Hintze, Philipp Hintze, Rainhard D. Findling, and Ren´eMayrhofer. A large-scale, long-term analysis of mobile device us- age characteristics. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT), 1(2):13:1–13:21, June 2017.

[6] Adam J. Oliner, Anand P. Iyer, Ion Stoica, EemilLagerspetz, and SasuTarkoma. Carat: Collaborative energy diagnosis for mobile devices. In Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, SenSys '13, pages 10:1–10:14, New York, NY, USA, 2013. ACM.

[7] Daniel T Wagner, Andrew Rice, and Alastair R Beresford. Device Analyzer: Large-scale mobile data collection. In Big Data Analytics workshop (in conjunction with ACM Sigmetrics 2013). ACM, 2014.

[8] Daniel T Wagner, Andrew Rice, and Alastair R Beresford. Device analyzer: Understanding smartphone usage. In International Confer- ence on Mobile and Ubiquitous Systems: Computing, Networking, and Services, pages 195–208. Springer, 2013.

[9] Andrew Rice and Simon Hay. Decomposing power measurements for mobile devices. In Proceedings of the 2010 IEEE International Conference on Pervasive Computing and Communications (PerCom), pages 70–78. IEEE, 2010.

[10] Dalal, S.; Jindal, U. Performance of integrated signature verification approach. In Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16–18 March 2016; pp. 3369–3373.

[11] Hossein Falaki, Ratul Mahajan, Srikanth Kandula, DimitriosLym- beropoulos, Ramesh Govindan, and Deborah Estrin. Diversity in smartphone usage. In Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys 2010), pages 179–194. ACM, 2010.

[12] Hien Truong, EemilLagerspetz, Petteri Nurmi, Adam Oliner, SasuTarkoma, and N. Asokan. The company you keep: Mobile malware infection rates and inexpensive risk indicators. In Proceedings of the 23rd International Conference on World Wide Web, WWW '14, pages 39 – 50. ACM, 2014.

[13] Qiang Xu, Jeffrey Erman, Alexandre Gerber, Zhuoqing Morley Mao, Jeffrey Pang, and ShobhaVenkataraman. Identifying diverse usage behaviors of smartphone apps. In Proceedings of the 11th ACM SIGCOMM Internet Measurement Conference, IMC '11, 2011.

[14] Pascal Welke, IonutAndone, Konrad Blaszkiewicz, and Alexander Markowetz. Differentiating Smartphone Users by App Usage. In Pro- ceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '16, 2016.

[15] Dalal, S., Seth, B., Jaglan, V. et al. An adaptive traffic routing approach toward load balancing and congestion control in Cloud–MANET ad hoc networks. Soft Comput (2022). https://doi.org/10.1007/s00500-022-07099-4

[16] EmmanouilKoukoumidis, Li-ShiuanPeh, and Margaret Rose Martonosi. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In Proceedings of the 9th interna- tional conference on Mobile systems, applications, and services, pages 127–140. ACM, 2011.

[17] SamuliHemminki, Petteri Nurmi, and SasuTarkoma. Accelerometer- based transportation mode detection on smartphones. In Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, page 13. ACM, 2013.

[18] Dalal, S., Agrawal, A., Dahiya, N., Verma, J. (2020). Software Process Improvement Assessment for Cloud Application Based on Fuzzy Analytical Hierarchy Process Method. In: , et al. Computational Science and Its Applications – ICCSA 2020. ICCSA 2020. Lecture Notes in Computer Science(), vol 12252. Springer, Cham. https://doi.org/10.1007/978-3-030-58811-3_70

[19] Valentin Radu, LitoKriara, and Mahesh K Marina. Pazl: A mobile crowdsensing based indoor wifi monitoring system. In Network and Service Management (CNSM), 2013 9th International Conference on, pages 75–83. IEEE, 2013.

[20] RuipengGao, Mingmin Zhao, Tao Ye, Fan Ye, Yizhou Wang, KaiguiBian, Tao Wang, and Xiaoming Li. Jigsaw: Indoor floor plan recon- struction via mobile

crowdsensing. In Proceedings of the 20th annual international conference on Mobile computing and networking, pages 249–260. ACM, 2014.

[21] Yohan Chon, Nicholas D Lane, Fan Li, Hojung Cha, and Feng Zhao. Automatically characterizing places with opportunistic crowdsensing using smartphones. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, pages 481–490. ACM, 2012.

[22] Erhard Rahm and Hong Hai Do. Data cleaning: Problems and current approaches. IEEE Bulletin of the Technical Committee on Data Engineering, 23(4):3–13, 2000.

[23] Ibrahim AbakerTargio Hashem, IbrarYaqoob, Nor BadrulAnuar, Salimah Mokhtar, AbdullahGani, and Samee Ullah Khan. The rise of "big data" on cloud computing: Review and open research issues. Information Systems, 47:98–115, 2015.

[24] Kyungseo Park, Eric Becker, Jyothi K Vinjumur, Zhengyi Le, and FilliaMakedon. Human behavioral detection and data cleaning in assisted living environment using wireless sensor networks. In Proceed- ings of the 2nd International Conference on PErvasive Technologies Related to Assistive Environments, page 7. ACM, 2009.

[25] Dalal, S., Dahiya, N., Jaglan, V. (2018). Efficient Tuning of COCOMO Model Cost Drivers Through Generalized Reduced Gradient (GRG) Nonlinear Optimization with Best-Fit Analysis. In: Saeed, K., Chaki, N., Pati, B., Bakshi, S., Mohapatra, D. (eds) Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing, vol 563. Springer, Singapore. https://doi.org/10.1007/978-981-10-6872-0_32

[26] James Williamson, Qi Liu, Fenglong Lu, Wyatt Mohrman, Kun Li, Robert Dick, and Li Shang. Data sensing and analysis: Challenges for wearables. In Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific, pages 136–141. IEEE, 2015.

[27] Malik, M., Nandal, R., Dalal, S., Jalglan, V., & Le, D. N. (2021). Driving pattern profiling and classification using deep learning. Intelligent Automation & Soft Computing, 28(3), 887-906.

[28] Jindal, U., Dalal, S., Rajesh, G., Sama, N. U., Jhanjhi, N. Z., & Humayun, M. (2021). An integrated approach on verification of signatures using multiple classifiers (SVM and Decision Tree): A multi-classification approach.

[29] Seth, B., Dalal, S., Le, D. N., Jaglan, V., Dahiya, N., Agrawal, A., ... & Verma, K. D. (2021). Secure Cloud Data Storage System Using Hybrid Paillier–Blowfish Algorithm. Computers, Materials & Continua, 67(1), 779-798.

[30] Vijarania, M., Dahiya, N., Dalal, S., & Jaglan, V. (2021). WSN Based Efficient Multi-Metric Routing for IoT Networks. In Green Internet of Things for Smart Cities (pp. 249-262). CRC Press.

[31] Goel, M., Hayat, A., Husain, A., & Dalal, S. (2021). Green-IoT (G-IoT) Architectures and Their Applications in the Smart City. In Green Internet of Things for Smart Cities (pp. 47-59). CRC Press.

[32] Chawla, N., & Dalal, S. (2021). Edge AI with Wearable IoT: A Review on Leveraging Edge Intelligence in Wearables for Smart Healthcare. Green Internet of Things for Smart Cities, 205-231.

[33] Dahiya, N., Dalal, S., & Jaglan, V. (2021). Efficient Green Solution for a Balanced Energy Consumption and Delay in the IoT-Fog-Cloud Computing. In Green Internet of Things for Smart Cities (pp. 113-123). CRC Press.

[34] Dahiya, N., Dalal, S., & Jaglan, V. (2021). Mobility Management in Green IoT. In Green Internet of Things for Smart Cities (pp. 125-134). CRC Press.

[35] Seth, B., Dalal, S., & Dahiya, N. (2021). Practical Implications of Green Internet of Things (G-IoT) for Smart Cities. In Green Internet of Things for Smart Cities (pp. 61-81). CRC Press.

[36] Dalal, S., Agrawal, A., Dahiya, N., & Verma, J. (2020, July). Software Process Improvement Assessment for Cloud Application Based on Fuzzy Analytical Hierarchy Process Method. In International Conference on Computational Science and Its Applications (pp. 989-1001). Springer, Cham.

[37] Seth, B., Dalal, S., Jaglan, V., Le, D. N., Mohan, S., & Srivastava, G. (2020). Integrating encryption techniques for secure data storage in the cloud. Transactions on Emerging Telecommunications Technologies.

[38] Hooda, M., & Shravankumar Bachu, P. (2020). Artificial Intelligence Technique for Detecting Bone Irregularity Using Fastai. In International Conference on Industrial Engineering and Operations Management Dubai, UAE (pp. 2392-2399).

[39] Arora, S., & Dalal, S. (2019). An optimized cloud architecture for integrity verification. Journal of Computational and Theoretical Nanoscience, 16(12), 5067-5072.

[40] Arora, S., & Dalal, S. (2019). Trust Evaluation Factors in Cloud Computing with Open Stack. Journal of Computational and Theoretical Nanoscience, 16(12), 5073-5077.

[41] Shakti Arora, S. (2019). DDoS Attacks Simulation in Cloud Computing Environment. International Journal of Innovative Technology and Exploring Engineering, 9(1), 414-417.

[42] Shakti Arora, S. (2019). Integrity Verification Mechanisms Adopted in Cloud Environment. International Journal of Engineering and Advanced Technology (IJEAT), 8, 1713-1717.

[43] Sudha, B., Dalal, S., & Srinivasan, K. (2019). Early Detection of Glaucoma Disease in Retinal Fundus Images Using Spatial FCM with Level Set Segmentation. International Journal of Engineering and Advanced Technology (IJEAT), 8(5C), 1342-1349.

[44] Sikri, A., Dalal, S., Singh, N. P., & Le, D. N. (2019). Mapping of e-Wallets With Features. Cyber Security in Parallel and Distributed Computing: Concepts, Techniques, Applications and Case Studies, 245-261.

[45] Seth, B., Dalal, S., & Kumar, R. (2019). Hybrid homomorphic encryption scheme for secure cloud data storage. In Recent Advances in Computational Intelligence (pp. 71-92). Springer, Cham.

[46] Seth, B., Dalal, S., & Kumar, R. (2019). Securing bioinformatics cloud for big data: Budding buzzword or a glance of the future. In Recent advances in computational intelligence (pp. 121-147). Springer, Cham.

[47] Jindal, U., & Dalal, S. (2019). A hybrid approach to authentication of signature using DTSVM. In Emerging Trends in Expert Applications and Security (pp. 327-335). Springer, Singapore.

[48] Le, D. N., Seth, B., & Dalal, S. (2018). A hybrid approach of secret sharing with fragmentation and encryption in cloud environment for securing outsourced medical database: a revolutionary approach. Journal of Cyber Security and Mobility, 7(4), 379-408.

[49] Sikri, A., Dalal, S., Singh, N. P., & Dahiya, N. (2018). Data Mining and its Various Concepts. Kalpa Publications in Engineering, 2, 95-102.

[50] Sameer Nagpal, S. (2018). Analysis of LrMu Power Algorithm in the Cloud Computing Environment using CloudSim Toolkit. International Journal of Research in Electronics and Computer Engineering (IJRECE), 6(3), 1175-1177.

[51] Nagpal, S., Dahiya, N., & Dalal, S. (2018). Comparative Analysis of the Power Consumption Techniques in the Cloud Computing Environment. Journal Homepage: http://www. ijmra.us, 8(8), 1.

[52] Kumar, N., Dalal, S., & Dahiya, N. (2018). Approach of Lion Optimization Algorithm for Efficient Load Balancing in Cloud Computing. Journal Homepage: http://www. ijmra.us, 8(8), 1.

[53] Sameer Nagpal, S. (2018). Comparison of Task Scheduling in Cloud Computing Using various Optimization Algorithms. Journal of Computational Information Systems, 14(4), 43-57.

[54] Arora, S., & Dalal, S. (2018). Hybrid algorithm designed for handling remote integrity check mechanism over dynamic cloud environment. International Journal of Engineering & Technology, 7(2.4), 161-164.

[55] Kukreja, S., & Dalal, S. (2018). Modified drosophila optimization algorithm for managing re-sources in cloud environment. International Journal of Engineering & Technology, 7(2.4), 165-169.

[56] Jindal, U., Dalal, S., & Dahiya, N. (2018). A combine approach of preprocessing in integrated signature verification (ISV). International Journal of Engineering & Technology, 7(1.2), 155-159.

[57] Nagpal, S., Dahiya, N., & Dalal, S. (2018). Comparison of Task Scheduling in Cloud Computing Using various Optimization Algorithms. Journal of Computational Information Systems ISSN, 1553-9105.

[58] Jindal, U., Dalal, S., & Dahiya, N. (2018). A combine approach of preprocessing in integrated signature verification (ISV). International Journal of Engineering & Technology, 7(1.2), 155-159

[59] Shakti Arora, S. (2018). Resolving problem of Trust context in Cloud Computing. International Journal of Engineering Research in Computer Science and Engineering (IJERCSE), 5(1), 138-142.

[60] Dalal, S., Dahiya, N., & Jaglan, V. (2018). Efficient tuning of COCOMO model cost drivers through generalized reduced gradient (GRG) nonlinear optimization with best-fit analysis. In Progress in Advanced Computing and Intelligent Engineering (pp. 347-354). Springer, Singapore

[61] Seth, B., & Dalal, S. (2018). Analytical assessment of security mechanisms of cloud environment. In Progress in Advanced Computing and Intelligent Engineering (pp. 211-220). Springer, Singapore.

[62] Kukreja, S., & Dalal, S. (2018). Performance analysis of cloud resource provisioning algorithms. In Progress in Advanced Computing and Intelligent Engineering (pp. 593-602). Springer, Singapore.

[63] Rani, U., Dalal, S., & Kumar, J. (2018). Optimizing performance of fuzzy decision support system with multiple parameter dependency for cloud provider evaluation. Int. J. Eng. Technol, 7(1.2), 61-65.

[64] Dahiya, N., Dalal, S., & Khatri, S. (2017). An Enhanced Bat Algorithm for Data Clustering Problems. International Journal of Advanced Research in Computer Science, 8(3).

[65] Dahiya, N., Dalal, S., & Khatri, S. (2017). Data clustering and its Application to numerical function optimization algorithm. International Journal of Advanced Research in Computer Science, 8(1).

[66] Arora, S., & Dalal, S. (2017). Adaptive Model For Integrity Verification In Cloud Computing System. International Journal of Advanced Research in Computer Science, 8(1), 233-236.

[67] Neeraj Dahiya, S. (2017). Numerical Function Optimization: Model, Procedure And Uses. International Journal of Engineering Science and Technology (IJEST), 9(4), 266-270.

[68] Dahiya, N., Dalal, S., & Khatri, S. (2016). Refinement with Image clustering using Self-Organizing Map and Numerical Function Optimization. International Journal of Computer Science and Information Security, 14(11), 909.

[69] Neeraj Dahiya, S. (2016). A Review on Numerical function optimization Algorithm and its Applications to Data Clustering & Classification. International Journal of Recent Research Aspects, 3(3), 115-121.

[70] Arora, S., & Dalal, S. (2016). Novel Approach of Integrity Verification in Dynamic Cloud Environment. International Journal of Computer Science and Information Security, 14(8), 207.

[71] Shawn R Jeffery, Gustavo Alonso, Michael J Franklin, Wei Hong, and Jennifer Widom. Declarative support for sensor data cleaning. Lecture Notes in Computer Science, 3968:83–100, 2006.

[72] Diane M Strong, Yang W Lee, and Richard Y Wang. Data quality in context. Communications of the ACM, 40(5):103–110, 1997.

[73] EimanElnahrawy and BadriNath. Cleaning and querying noisy sensors. In Proceedings of the 2nd

ACM international conference on Wireless sensor networks and applications, pages 78–87. ACM, 2003.

[74] S. Arora and S. Dalal, "Hybrid algorithm designed for handling remote integrity check mechanism over dynamic cloud environment", International Journal of Engineering & Technology, pp. 161-164, 2018.

[75] DimitriosLymberopoulos, Athanasios Bamis, and Andreas Savvides. Extracting spatiotemporal human activity patterns in assisted living using a home sensor network. Universal Access in the Information Society, 10(2):125–138, 2011.

