Journal of Theoretical and Applied Information Technology
31st August 2023. Vol.101. No 16

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

PRAGMATIC INVESTIGATIONS TO SMART DUSTS LOCATION APPRAISAL PRECISELY USING MACHINE LEARNING

MR. TATA BALAJI[1] , KURRA UPENDRA CHOWDARY[2], DR. P VENU MADHAV[3] . [4]DR.A GEETHA DEVI MRS.T. MAHALAKSHMI[5] DR.SURYA PRASADA RAO BORRA[6] N.JAYA[7]

[1]Sr.Asst Professor, Dept of ECE, PVP Siddhartha Institute of Technology, Vijayawada India
[2]Assoc Professor, Dept of ECE, R.V.R& J.C.College of Engineering, Guntur, AP India
[3]Asst Professor, Dept of ECE, PVP Siddhartha Institute of Technology: Vijayawada India
[4]Assoc Professor, Dept of ECE, PVP Siddhartha Institute of Technology, Vijayawada India
[5]Asst. Professor, Dept of ECE, PVP Siddhartha Institute of Technology: Vijayawada India
[6]Associate Professor, Dept of ECE, PVP Siddhartha Institute of Technology India
[7]Department of EIE, Faculty of Engineering and Technology, Annamalai University, Chidambaram, India

E-mail: balajitata@pvpsiddhartha.ac.in, kupendra@rvrjc.ac.in, venumadhav@pvpsiddhartha.ac.in, geetha.agd@gmail.com, dasari.maha@pvpsiddhartha.ac.in, suryaborra1679@gmail.com, jayanavaneethan@rediffmail.com

ABSTRACT

Rough terrain that is difficult or impossible to access does not lend itself well to traditional Wireless Sensor Networks (WSNs). Smart dust is a technology that gathers remote sensing data from harsh terrain by utilising a network of numerous microscopic sensors. The small sensors are dispersed in large numbers across difficult terrains using airborne distribution from drones or aeroplanes, eliminating the need for manual placement. Although it is clear that this technology can be applied to a wide range of remote sensing applications, the small size of smart dusts essentially precludes the integration of complex circuitry on tiny sensors. This poses a number of challenges, one of which is locating the smart dusts. In order to pinpoint the precise location of events detected by the smart dusts, this study suggests a localization algorithm. General regression neural network is used in the method to forecast the locations. Because real smart dusts aren't readily available, we created a simulator to assess the proposed method's accuracy when used to monitor forest fires. The simulation trials indicate that the method is reasonably accurate.

Keywords: *Smart Dust, Machine Learning Algorithms, IOT, Sensor Applications*

1. INTRODUCTION

In recent years, studies on wireless sensor networks (WSNs) have demonstrated their efficacy for a variety of remote sensing applications, including precision agriculture [1-3], logistics carrier monitoring [4, 5], industrial monitoring and control [6, 7], monitoring of underground coal mines [8, 9], monitoring of seismic activity [10], monitoring of the marine environment [11], and so on. The fact that WSN technology is ad hoc and simple to administer and deploy is its main benefit. However, because WSN deployment involves assistance from people, it is not appropriate for rocky terrain that is challenging or impossible for humans to access. In this study, we employ cutting-edge technology (smart dust) to track remote sensing uses for unforgiving terrains [12–14]. We put smart dust to the test for detecting forest fires. The technology known as "smart dust" relies on the cooperation of several small microelectromechanical sensors (MEMS) [15–17]. These small sensors can pick up on a wide range of properties, including pressure, humidity, temperature, vibration, acceleration, and light [14, 17, 18]. The technology for detecting and wireless transmission is built into each tiny sensor. The autonomous power source for smart dust can come from thick-film batteries or solar energy. In a tiny sensor with a footprint of just a few millimeters, all these capabilities are condensed [19–21]. A smart dust's architecture is depicted in Figure 1.

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

Emerging technology is Smart Dust. One of the upcoming technologies over the next ten years, according to the Gartner hype cycle for 2018, is smart dust. This technique has drawn a lot of interest from the scientific community in recent years due to the fact that it may be used in a variety of remote sensing applications to monitor difficult terrains, including forests, frozen areas, mountains, the sea, space, planets, etc. [22–25]. Researchers studying the environment can use this technology, for example, to count the number of pollutants, carbon, or any other chemical component in the air. Using this technology, the agricultural industry can keep tabs on soil quality, the ideal time to harvest crops, and other sensory information that is good for plant growth [26]. Geologists can gain by keeping an eye on seismic or earthquake activity and how it affects the design of bridges and other structures. This technique can be used by biologists to track the movements of little insects or other wild creatures in order to learn more about their natural surroundings. This technique can also be used by the military to track troops or find radioactivity or dangerous chemical substances in the atmosphere [25, 27].

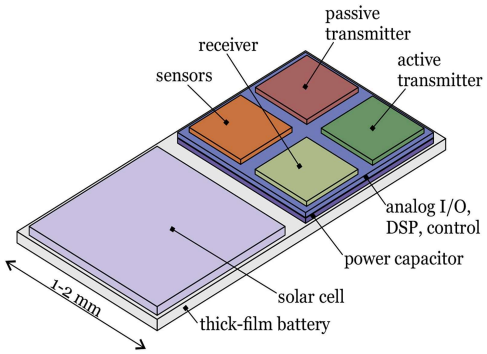


FIGURE 1: PROPOSES ARCHITECTURE FOR SMART DUST.

When smart Dusts are dispersed in huge numbers on a monitoring zone without being manually located, aerial dispersion utilising drones or planes is the optimum way for scattering sensors throughout a specific area [12]. Because these sensors may stay in the environment like dust, they may be difficult to find and remove after deployment. The smart dusts detect the predefined events and pass them from sensor to sensor until they reach the base stations. In order to perform a task, the base stations send events to the cloud for processing or analytics. The fundamental functions of smart dusts are to detect events, briefly store them in memory, and wirelessly broadcast them to adjacent sensors or base stations. Despite the fact that it is clear that a variety of distant sensing applications could benefit from this technology, there are numerous challenges involved in distributing a large number of smart dust sensors throughout a specific monitoring zone, including sensor localization, device failure, and bottleneck [28]. The main focus of localization research in WSNs is locating the sensors [29, 30]. Since sensors are used to monitor the existence of preprogramed events, pinpointing the precise position of an event is essential for many applications, such as tracking troop movements, wild animals, and forest fires. The tiny size of smart dust, on the other hand, results in a variety of constraints, including those related to energy usage, wireless communication, and scalability, which fundamentally distinguishes this technology from others such as WSNs or the internet of things (IoT) [31]. For example, the majority of IoT sensors on the market now are the size of matchboxes, providing adequate space for an embedded CPU with high performance, a radio transceiver with a longer communication range of tens of meters, a modest amount of Memory, and a shared bandwidth of tens of kilobits. These sensors are too costly, too big, and have a short battery life. Thus, these are not suitable for applications that require energy-efficient sensors that should last for years without a charge or battery replacement. Moreover, the sensors for these applications need to be small, light, and inexpensive enough to be easily sprinkled on a monitoring area as well as easy to combine with coatings and paint.

Because of the energy, size, and cost constraints of this technology, smart dust sensors have much lower processing, communication, and storage capabilities than WSNs or IoT devices [31]. Moreover, the smart dust cannot locate itself via GPS or radio waves because those technologies would demand additional resources that would be unsuitable in terms of space, money, and energy. Future hardware advances of this technology will likely focus more on size reduction, energy optimization, and communication than on equipping the smart dusts with cutting-edge technology because of the technology's specific uses [32]. There have lately been several range-based and range-free sensor localization approaches published in the context of WSNs and IoT domains [29, 30, 33, 34]. These systems require sensors to be equipped with long-range wireless chips in order for them to instantly relay their signals to anchor (base) sensors for the purpose of detecting the position of sensors. These localization algorithms,

however, cannot be used directly on smart dusts because to resource limitations (long-range wireless technology is not available on smart dusts) [31]. As a result, new localization strategies that are effective in the context of smart dust are required. We take these constraints into consideration in this study and propose a unique localization technique for smart dust that efficiently pinpoints the location of events perceived by the smart dusts. The proposed technique pinpoints the location of occurrences by using signal timestamps (time of arrival) received from base stations.

The approach works as follows. The monitoring zone is split into n cells to calculate the location of an incident. The approach assumes that each cell has an equal number of distributed smart dusts. The monitoring zone is surrounded by m manually positioned base stations. These base stations can report their precise locations via GPS and have cutting-edge hardware and software characteristics. Figure 2 provides an illustration of how the system uses base stations and smart dusts to monitor forest fires. The smart dust signals to nearby smart dusts when it detects an event. The signal is retransmitted to more neighbors by the nearby smart dusts. With this method, the perceived event travels from source smart dusts to base stations. Because of the positioning of the base stations across multiple geographies and the extent of the monitoring zone, each base station acquires unique (time of arrival) timestamps for distinct sensed events taking place at various geographic locations. We were able to generate a large number of samples for training a machine learning classifier utilising this concept by simulating multiple events across various geographic locations inside a simulated monitoring region. Finally, using a neural network and training data, we created a machine learning (ML) classifier. When the ML classifier is ready, the system uses the time of arrival of the signals base stations received to accurately predict the position of each sensed event. The suggested method can scale up to very large networks.

2. ALLIED EXERTION

The tiny size and limited resources on the sensor provide various challenges for smart dust technologies. Park et al. [12] proposed a hierarchical layered system for smart dust dispersal. According to the authors, the proposed approach reduces the transmission bottleneck caused by many smart dusts and dynamically creates plan partitions for a particular workload. The three layers of the suggested architecture are where the smart dust and gadgets are placed. Smart dusts are present in the top layer and are utilized to monitor their surroundings. The smart dusts' short communication range prevents them from sending sensed events across great distances. Dust relay sensors are located in the second layer. Despite having a small number, these sensors have sufficient computation and transmission power to process and communicate data over a considerable distance to the higher layer. A pool control node and a smart IoT server with several processing nodes are both present in the third layer. Processing nodes are responsible for processing data from relay dust sensors. The workload on the processing nodes is dynamically distributed by the pool control node. The authors of [16] employed this hierarchically tiered design to keep track of the climate data for a fictitious remote area.

The transmission technique was created by Park et al. [28] to effectively transmit events noticed by the smart dusts to IoT servers. Sensible data is divided into two kinds in the recommended procedure: ordinary sensed data and sensible data of a certain urgency. The data in the urgent class needs to be sent to the IoT server right away, while the data in the regular class can be sent with a small delay because of the transmission of the data in the urgent class. To further safeguard the data of the typical class, the authors used block chain technology. The processing of data in the urgent category is forwarded urgently to an IoT server. When there are no data accessible for the urgent condition, the data of the regular detected class is first recorded on the block chain ledger and then transmitted.

A use of smart dust technology for surveillance was suggested by Mohan et al. [27]. They talked on the role that smart dust technology can play in terrorism border surveillance. A smart court system using smart dust technology was presented by Jain et al. [35]. Throughout their work, they covered the topic of how technology may be employed in smart cities to identify and lessen crime. They also talked about new uses for smart dust technology, including meta security, privacy protection systems, and preventing the transmission of bacteria and fungus that cause disease.

Romer et al. [31] recommended employing a localization approach to locate smart dusts. To detect their location, the smart dusts employ laser patterns sent to them by an infrastructure device. However, this strategy is ineffective for monitoring places with uneven surfaces (such as woodlands,

mountains, or planets), since smart dusts cannot perceive the infrastructure device's laser patterns owing to barriers on uneven surfaces. The technique we propose differs from that described in the published paper. We built a machine learning classifier utilising signal timestamps (time of arrival) obtained from base stations to properly estimate the position of smart dusts. Our approach is extremely versatile, since it may be employed over large areas of difficult-to-access mountainous terrain.

3 APPROXIMATING LOCALITY OF SMART DUSTS

The recommended technique aims to predict event locations reasonably precisely. The recommended system should also be able to operate within the constraints of power, cost, and communication range while covering a substantial monitoring zone. The smart dust is a small device that collaborates with other smart dusts to achieve its objective. Due to their limited computer and communication capabilities, these small devices are unable to transmit detected events across great distances. An RFID-based system and (ii) a light source-based system are the two main techniques for allowing wireless communication for smart dusts [32]. Compared to optical transceivers, RFID-based communication is more appealing since it may be used to monitor areas that are difficult to access because it does not require a direct line of sight for communication. Smart dust that uses RFID is exemplified by Hitachi's smart tag technology. The Hitachi minuscule chip is an RFID chip that can be used for wireless sensing and has a miniscule CPU and ROM of size 128b. All of these features are built into a chip with a 0.15 x 0.15 mm surface area and a 75 m height.

Another layer of cutting-edge sensors known as base stations in the monitoring region convey discovered events over a long distance or to a cloud server [12]. These base stations are manually positioned along the perimeter of the monitoring region. The base stations are outfitted with cutting-edge technology and software, and they may employ GPS to relay their precise positions to a cloud server for extra predictive analytics and decision-making. The following illustrates how the smart dust system for remote sensing works. After sensing a pre-programmed event, the smart dust communicates the perceived event to other smart dusts within its RFID communication range. Upon signal receipt, surrounding smart dusts retransmit the signal to further nearby smart dusts. In this way, the perceived event travels from the source smart dusts to the base stations. Figure 2 displays the architecture of smart dust-based remote sensing for identifying forest fires. Each smart dust comprises a threshold module that regulates the quantity of data transferred, conserves energy, and reduces bottleneck [16]. When the difference between the current event and the preceding event is less than a predefined threshold, the smart dust employs the threshold module to prevent the observed event from being sent to neighboring sensors. When an incident in the monitoring region is identified, the smart dusts transmit a signal to the base stations. Each base station receives a unique signal timestamp for every detected event that takes place at a unique point within the monitoring region due to the varying geographical placements of the base stations.

Figure 3 illustrates an illustration of how base sensors obtain varied timestamps for three events taking place in different places. At random locations throughout the monitoring area, a varying number of smart dusts are scattered. There is a few-meter signal transmission range for every intelligent dust particle. Five base stations are spread out evenly over the monitoring area. Compared to other sensors, the base station I detects the event A faster. This is so because sensor I is closest to the site of event A compared to the other sensors. The timestamps acquired at other lower-level sensors for event A differ depending on their proximity to event A. Similarly, the base station j detects event B sooner than other sensors since it is closest to it. We used this idea, as well as a large number of training data, to develop a machine learning (ML) classifier using a neural network by simulating numerous events at various places inside the monitoring zone. After the ML classifier is ready, the base stations transmit the timestamps of detected events to the cloud server. The cloud server aggregates the timestamps into a multidimensional vector and feeds it to an ML classifier to predict the location of observed events.

Using multi-output regression, the system learns an ML classifier [36–37]. The goal of multi-output regression, in contrast to normal regression, which aims to predict a single numerical value, is to anticipate the numerical outputs of two or more numerical characteristics. In this case, they are the geographic coordinates of observed occurrences. A multi-output regression predictor can be trained using many conventional ML classifiers. Decision trees or ensembles of decision tresses, for instance, can be utilized for this. Decision trees, however, are

not appropriate when the training samples include input and output attribute relationships that are highly structured. Multi-output regression is supported by neural networks employing GRNNs (generic regression neural networks). It's possible to use it to train a continuous function that can identify a more agreeable connection between input and output. The multi-output regression is learned by the GRNN by explicitly changing the numeric output characteristics on the output layer. The classifier is trained by the GRNN using kernel regression. It is one sort of radial basis function (RBF) network and is based on the conventional statistical method [38]. Recent research on GRNN has shown that it is a training-efficient network that offers relatively high accuracy even when it is trained with a small sample size [39–40].

To understand the GRNN's learning procedure, assume there is a training sample with input and output vectors. The input vector has m independent input characteristics Ki = [k1, k2,..., km], but the output vector has k dependent features Mi = [m1, m2,...mk]. If enough training samples are submitted to the network, the GRNN may effectively learn the regression surface (linear or non-linear) and predict the values of the dependent outputs of a new unknown sample Kj. The following are the steps in GRNN training:

$$E[\boldsymbol{M}|\boldsymbol{K}] = \frac{\int_{-\infty}^{+\infty} \boldsymbol{K f(M,K) dK}}{\int_{-\infty}^{+\infty} \boldsymbol{f(MK) dK}} \quad \textbf{--- (1)}$$

The predicted value of the attribute M is represented by the E[M|K] given an input vector K from training samples n. The probability density of the joining of K and Y is represented by the function f(M, K).

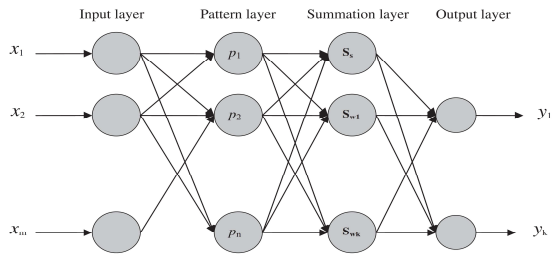


Figure 2: GRNN (General Regression Neural Network) Architecture

The GRNN architecture consists of four levels. Output layer, pattern layer, summation layer, and input layer are these. The GRNN's architecture is depicted in Figure 4. The total number of characteristics in the input vector Ki is equal to the number of nodes (neurons) in the input layer. Each node of the input vector represents a distinct attribute. Each input layer node transmits its data value to every node in the pattern layer. The total number of training samples is represented by the number of nodes on the pattern layer. The pattern layer receives the values of the input attributes in order to comprehend the connection between nodes of the input layer and the (proper reaction) nodes of the output layer. This map between both the input layer and the pattern layer transforms the input space into the pattern space in a nonlinear manner. The following is a description of the pattern layer pi's Gaussian function:

$$\boldsymbol{p_i = exp\left[-\frac{(k-k_i)^T(k-k_i)}{2\sigma^2}\right], \quad i = 1, 2, \dots, n} \quad (2)$$

The parameter for smoothing is σ. k is a network input variable. A sample from the training set for the pattern layer's node I is called Ki.

There are two different types of summing functions in the summation layer. Simple summation and weighted summation are these. A simple summation is S_s. With an interconnection weight of 1, it adds up the values received from the pattern layer nodes. S_w uses the interconnection weight of w to calculate the weighted sum of the pattern layer nodes using t. The definition of the summation functions is:

$$\boldsymbol{S_s = \sum_{t=1}^{n} p_t} \quad (3)$$

$$\boldsymbol{S_w = \sum_{t=1}^{n} w_t p_t} \quad (4)$$

The weight of the pattern layer node that is connected to the summation layer is indicated by the w_t.

The output layer is the bottom layer, and it has exactly as many nodes as there are dependent characteristics in all of the training samples. Two nodes of the output layer can be used to specify the two numeric attributes of the geographical co-ordinate's prediction task. The output layer receives the calculated values from the summing layer. The node's output is calculated as follows at the output layer:

$$\boldsymbol{M_o = \frac{S_s}{S_w}, \quad o = 1 \dots k} \quad (5)$$

The spread parameter is critical in training the GRNN for the optimum prediction accuracy [41]. Training the GRNN with a lesser value of σ, for example, can result in a localized regression prediction. In this situation, training samples that are very close to the prediction sample's neighbour only contribute to the prediction of final output. Training the GRNN with a bigger value of σ, on the other hand, can result in globalized regression prediction, which incorporates all of the samples for predicting the value of the output node. The projected value of the output node in this scenario is fairly close to the average value of the dependent attribute across all samples of the training dataset.

Because GRNNs are stochastic, it is critical to test the network several times using the same training data. For this aim, we use k-fold cross validation with 10 folds. To do this, we randomly split the training samples into five groups, with an equal number of samples in each group. Lastly, we use cross validation to randomly choose four groups to discover an ideal value for the spread parameter. The first set is comprised of the four groups that were chosen. The remaining group is then included in the construction of a second set, which is then used to assess the prediction accuracy. In order to train the GRNN using cross validation, we build the model using a random value of the spread parameter (σ) on a random three groups from the first batch. The model is assessed on the group from the second set after training. If the GRNN model has the highest prediction accuracy on the test partition, it is output as the result. If the GRNN model does not produce the best prediction accuracy, the system retrains the model with a new value (σi) for the speared parameter (σ). The following rule produces the new value of (σ).

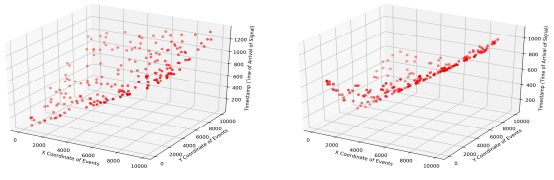
$$\sigma_i = \sigma_o + (i \times \eta) \quad (6)$$

The initial spread parameter is σo. In the experiment, the σo is set to a low value of 0.1. The learning ratio is indicated by the η. We utilize the with 0.1 in the experiments. When the cross validation's mean square error is less than the target error, e0, training is terminated. In experiments, the e0 with 0.01 is used. When to stop training to avoid over-fitting is decided using a different criterion (maximum number of iterations). We do the trials up to a maximum of 20 times. When the number of training phase iterations is greater than 20, the system randomly selects new sets for training, testing, and cross validation.

4. EXPERIMENTS

Unfortunately, true smart dust hardware is currently unavailable. We created a simulator to evaluate the efficacy of the proposed study for tracking forest fires, which is just one way used to illustrate the practicality of the proposed strategy. Our ultimate objective is to implement a practical project that uses smart dusts to monitor forest fires. The tests were carried out using an Intel Core i7-7th generation Processor with a 2.11 GHz clock speed and an 8 GB main memory. We constructed a rectangular-shaped simulated monitoring region with a width and length of 10 kilometers. One kilometer wide by one-kilometer-long partitions were used to split the monitoring area into 100 parts. On the edge of the monitoring area, sixteen base stations were positioned with equal spacing between them. Smart dusts were applied to partitions with various densities using simulation. Equal numbers of smart dusts were distributed randomly onto each partition, one for each partition. Each smart dust particle could relay the signal of the sensed event over a 25-meter-radius. A random setting of 1 nanosecond was used for the time it takes for a signal to travel from a smart dust to the other or from a base station to another. For the simulation, stochastic network elements like signal loss and node failure were taken into account.

The simulator creates fire incidents at random in the monitoring zones. If a fire occurs within the range of the smart dusts, they detect it and transmit a signal to the base stations. The base stations collect signal timestamps and send them to a cloud server, which aggregates them in to the training set to train a GRNN. To alleviate the transmission bottleneck, each smart dust has an event detection threshold. When the value of an observed event exceeds the threshold, the smart dust generates a signal and sends it to nearby smart dusts. If there are enough generated samples, the system trains the GRNN using the manner outlined in section 3



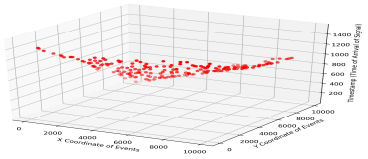


Figure 3: Base Station Performance At The Left Edge In Relation To Detected Events Occurring At Different Locations In The Monitoring Zone.

In the second component, we examined the suggested method's accuracy in the event that some base stations were unable to receive timestamps from the smart dusts due to the smart dusts being believed to be defective. We sought to assess the ML predictor's performance in this domain if the unknown input samples had missing values, because base station data serve as dimensions of an input sample. Base station values with varying percentages were deleted at random in order to conduct the experiments. Using percentages of 25%, 50%, 75%, and 100%, we eliminated the base station values from every edge. Figure 8 depicts the GRNN predictor's accuracy for different percentages of missing variables. The accuracy of the GRNN predictor was greatly affected by more than 50% of base stations having missing values. This shows that even if a few smart dusts were malfunctioning and unable to communicate felt events to base stations, the suggested approach can still reasonably estimate the event's location

The results reveal that when events occur close to the base station, the values of signal arrival timestamps are smaller than when events occur far away from the base station

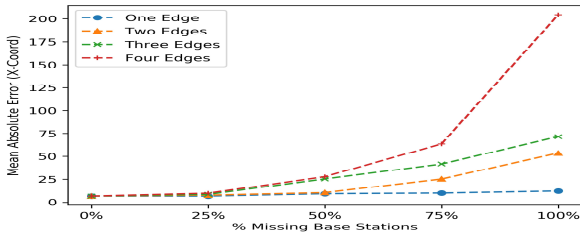


Figure 4. GRNN Effectiveness On Missing Base Station Values.

5. CONCLUSION

Smart dust is a revolutionary technology with several applications for distant sensing in difficult-to-reach areas. Yet, the small size of smart dust severely limits the integration of cutting-edge IoT technology on miniscule sensors. One of the difficulties was locating smart dusts. The paper describes a strategy for finding events observed by smart dusts. Using a generic regression neural network, the approach predictions the locations. Because genuine smart dusts were not widely accessible, we created a simulator to evaluate the effectiveness of the proposed technique for the application of detecting forest fires. While there are different techniques to assessing the accuracy of the proposed methodology, our main goal is to analyse the accuracy using real smart dusts across a vast monitoring zone. According to tests on a specially constructed simulator, the technique offers an acceptable degree of accuracy. To assess the strategy's success, we did experiments using a variety of factors. When about 60% of base sensors acquire event timestamps from smart dusts, the suggested strategy is 60% more successful. A GRNN was used to train the current prediction model (General Regression Neural Network). The GRNN has a performance advantage, but it takes longer to train. Further research may be required to develop a predictive classifier that may expedite computing while improving accuracy.

REFERENCES

[1] P. Sanjeevi, S. Prasanna, B. Sivakumar, G. Gunasekaran, I. Alagiri *et al.*, "Precision agriculture and farming using internet of things based on wireless sensor network," *Emerging Telecommunications Technologies*, vol. 31, no.12, pp. 1-12, 2020.

[2] C. Jamroen, P. Komkum, C. Fongkerd and W. Krongpha, "An intelligent irrigation scheduling system using low-cost wireless sensor network toward sustainable and precision agriculture," *IEEE Access*, vol. 8, no.1, pp. 172756–172769, 2020.

[3] L. Garcia, L. Parra, J. M. Jimenez, J. Lloret and P. Lorenz, "IoT-based smart irrigation systems: An overview on the recent trends on sensors and IoT systems for irrigation in precision agriculture," *Sensors*, vol. 20, no.4, 2020.

[4] W. Wang, "A remote monitoring system of logistics carrier based on wireless sensor network," *International Journal of Online and Biomedical Engineering*, vol. 14, no.1, pages 4–16, 2018.

[5] J. A. Luis, J. A. G. Galan, F. Gomez-Bravo, M. Sanchez-Raya, J. A. Espigado *et al.,* "An efficient wireless sensor network for industrial monitoring and control," *Sensors*, vol. 18, no.1, pp 182-197, 2018.

[6] W. Chen and X. Wang, "Coal mine safety intelligent monitoring based on wireless sensor network," *IEEE Sensors Journal*, 2020.

[7] L. Muduli, D. P. Mishra and P. K. Jana, "Application of wireless sensor network for environmental monitoring in underground coal mines: A systematic review," *Journal of Network and Computer Applications.*, vol. 106, no.1, pp. 48–67, 2018.

[8] H. Noureddine and K. Bouabdellah, "Using wireless multimedia sensor networks to enhance early forest fire detection," *International Journal of Distributed Systems and Technologies*, vol. 11, no.3, pp, 1–21, 2020.

[9] N. Varela, D.-M. Jorge L, A. Ospino and N. A. L. Zelaya, "Wireless sensor network for forest fire detection," *Procedia Computer Science*, vol. 175, no.1, pp. 435–440, 2020.

[10] K. K. Khedo, Y. Bissessur and D. S. Goolaub, "An inland wireless sensor network system for monitoring seismic activity," *Future Generation Computer Systems*, vol. 105, no.1, pp. 520–532, 2020.

[11] G. Xu, Y. Shi, X. Sun and W. Shen, "Internet of things in marine environment monitoring: A review," *Sensors*, vol. 19, no.7, pp. 1711–1732, 2019.

[12] J. Park and K. Park, "A dynamic plane prediction method using the extended frame in smart dust IoT environments," *Sensors*, vol. 20, no.5, pp. 1364–1380, 2020.

[13] L. Niccolai, M. Bassetto, A. A. Quarta and G. Mengali, "A review of smart dust architecture, dynamics, and mission applications," *Progress in Aerospace Sciences*, vol. 106, no.1, pp. 1–14, 2019.

[14] B. Warneke, M. Last, B. Liebowitz and K. S. J. Pister, "Smart dust: Communicating with a cubic-millimeter computer," *Computer*, vol. 34, no.1, pp. 44–51, 2001.

[15] M. Holler, B. van Giffen, L. Barth and R. Fuchs, "Smart dust for smart(er) industrial product-service-systems: Three strategies and their application," in *Proceedings of Smart Services Summit*, pp. 15–20, 2021.

[16] J. Park and K. Park, "Construction of a remote monitoring system in smart dust environment," *Journal of Information Processing Systems*, vol. 16, no.3, pp. 733–741, 2020.

[17] J. M. Kahn, R. H. Katz and K. S. J. Pister, "Emerging challenges: Mobile networking for smart dust," *Journal of Communications and Networks*, vol. 2, no.3, pp. 188–196, 2000.

[18] J. M. Kahn, R. H. Katz and K. S. J. Pister, "Next century challenges: Mobile networking for smart dust," in *Proceedings of Fifth Annual ACM/IEEE International Conference on Mobile Computing and Networking, Seattle, Washington, USA,* pp. 271–278, August 15-19, 1999.

[19] B. Atwood, B. Warneke and K. Pister, "Preliminary circuits for smart dust," In *Proceedings of 2000 Southwest Symposium on Mixed-Signal Design*, pp. 87–92, 2000.

[20] Z. Karakehayov, "Zero-power design for smart dust networks," In *Proceedings of First International IEEE Symposium Intelligent Systems*, vol. 1, pp. 302–305, 2002.

[21] T. Watteyne, "Crystal-free architectures for smart dust and the industrial IoT," in *Proceedings of 2020 7th International Conference on Internet of Things: Systems, Management and Security*, pp. 1–1, 2020.

[22] M. Holler, J. Haarmann, B. van Giffen and A. G. Frank, "Smart dust in the industrial economic sector - on application cases in product lifecycle management," in *Proceedings of IFIP International Conference on Product Lifecycle Management,* pp. 165–175, 2020.

[23] M. Holler, C. Dremel, B. van Giffen and R. Fuchs, "Smart dust und micro robots im industriellen sector," *HMD Praxis der Wirtschaftsinformatik.*, vol. 57, no. 1, pages 1239–1250, 2020.

[24] R. M. Aileni, G. Suciu, M. Serrano, R. Maheswar, C. A. V. Sakuyama *et al.,* "The perspective of smart dust mesh based on IoEE for safety and security in the smart cities," in *Integration of WSN and IoT for Smart Cities, Book Chapter*, pp. 151-179, 2020.

[25] S. Sathyan and S. R. Pulari, "A deeper insight on developments and real-time applications of smart dust particle sensor technology," in *Computational Vision and Bio Inspired Computing*, *Book Chapter*, pp. 193–204, 2018.

[26] G. N. Rameshaiah, Jpallavi and S. Shabnam, "Nano fertilizers and nano sensors - an attempt for developing smart agriculture," *International Journal of Engineering*

Research and General Science, vol. 3, no.1, 2015.

[27] C. Mohan and S. Arulselvi, "Smartdust network for tactical border surveillance using multiple signatures," *IOSR Journal of Electronics and Communication Engineering*, vol. 5, no.1, pp. 1–10, 2013.

[28] J. Park and K. Park, "A two-class data transmission method using a lightweight blockchain structure for secure smart dust IoT environments," *Sensors*, vol. 20, no.21, pp. 6078–6095, 2020.

[29] A. K. Paul and T. Sato, "Localization in wireless sensor networks: A survey on algorithms, measurement techniques, applications and challenges," *Journal of Sensor and Actuator Networks*, vol. 6, no.4, pp. 24–47, 2017.

[30] F. Mekelleche and H. Haffaf, "Classification and comparison of range-based localization techniques in wireless sensor networks," *Journal of Communications*, vol. 12, no.4, pp. 221–227, 2017.

[31] K. Romer, "Tracking real-world phenomena with smart dust," In *Proceedings of First European Workshop, EWSN 2004, Springer, Berlin, Germany,* pp. 28–43, *January 19-21,* 2004.

[32] D. Sadana, N. Li, S. Bedell and G. S. Shahidi, "Smart dust and internet of things (IoT): Progress & challenges," *Journal of Lasers, Optics & Photonics*, vol. 4, no.1, 2017.

[33] H. Xu, "Semi-supervised manifold learning based on polynomial mapping for localization in wireless sensor networks," *Signal Processing*, vol. 172, no.1, pp. 107570, 2020.

[34] K. Akhil and S. Sinha, "Self-localization in large scale wireless sensor network using machine learning," in *Proceedings of 2020 International Conference on Emerging Trends in Information Technology and Engineering*, pp. 1–5, Vellore, India, 2020.

[35] S. K. Jain and N. Kesswani, "Smart judiciary system: A smart dust based IoT application," in *Proceedings of Emerging Technologies in Computer Engineering: Microservices in Big Data Analytics*, pp. 128–140, Singapore, 2019.

[36] O. Polat and T. Yildirim, "Hand geometry identification without feature extraction by general regression neural network," *Expert Systems with Applications*, vol. 34, no.2, pp. 845–849, 2008.

[37] D. Specht, "A general regression neural network," *IEEE Transactions on Neural Networks*, vol. 2, no.6, pp. 568–576, 1991.

[38] H. B. Celikoglu, "Application of radial basis function and generalized regression neural networks in non-linear utility function specification for travel mode choice modelling," *Mathematical and Computer Modelling*, vol. 44, no.7, pp. 640–658, 2006.

[39] A. H. Manek and P. K. Singh, "Comparative study of neural network architectures for rainfall prediction," in *Proceedings of 2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, pp. 171–174, 2016.

[40] Y. Ni and M. Li, "Wind pressure data reconstruction using neural network techniques: A comparison between BPNN and GRNN," *Measurement*, vol. 88, no.1, pages 468–476, 2016.

[41] J. Liu, W. Bao, L. Shi, B. Q. Zuo and W. Gao, "General regression neural network for prediction of sound absorption coefficients of sandwich structure nonwoven absorbers," *Applied Acoustics*, vol. 76, no.1, pp. 128–137, 2014.

[42] D. Kim and S. Park, "Reinforcement Learning-Based Dynamic Adaptation Planning Method for Architecture-Based Self-Managed Software," *in Software Engineering for Adaptive and Self-Managing Systems*, 2009. SEAMS'09. ICSE Workshop on, 2009.

[43] H. N. Ho and E. Lee, "Model-Based Reinforcement Learning Approach for Planning in Self-Adaptive Software System," *in Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication,* 2015.

[44] Elkhodary, N. Esfahani and S. Malek, "FUSION: A Framework for Engineering Self-Tuning Self-Adaptive Software Systems," *in Proceedings of the eighteenth ACM SIGSOFT international symposium on Foundations of software engineering,* 2010.

[45] N. Esfahani, A. Elkhodary and S. Malek, "A Learning-Based Framework for Engineering Feature-Oriented Self-Adaptive Software Systems," *Software Engineering, IEEE Transactions on*, vol. 39, pp. 1467-1493, 2013.