

Article

Support Vector Machine-Based Energy Efficient Management of UAV Locations for Aerial Monitoring of Crops over Large Agriculture Lands

Mohammed Al-Naeem ^{1,*} , M M Hafizur Rahman ¹ , Anuradha Banerjee ² and Abu Sufian ³ 

¹ Department of Computer Networks & Communications, CCSIT, King Faisal University, Al Ahsa 31982, Saudi Arabia

² Department of Computer Applications, Kalyani Government Engineering College, Maulana Abul Kalam Azad University of Technology, Kalyani 741235, India

³ Department of Computer Science, University of Gour Banga, English Bazar 732103, India

* Correspondence: naeem@kfu.edu.sa

Abstract: Crop monitoring and smart spraying have become indispensable parts of precision agriculture where unmanned aerial vehicles (UAVs) play a lead role. In particular, in large agricultural fields, aerial monitoring is a sustainable solution provided it can be performed in an energy-efficient manner. The existing literature points out that the research on precision agriculture using UAVs is still very minimal. In this article, we propose a support vector machine (SVM)-based UAV location management technique where UAVs change position over various portions or regions of a large agricultural field so that crops are properly monitored in an energy-efficient manner. Whenever a processing request is generated from any sensor in a part of the field, the UAV investigates with an SVM to decide whether to move on to the center of that field based on various parameters or characteristics such as region-id, packet-id, time of day, waiting times of the packets, the average waiting time of others within a predefined time window, location of the UAV, residual energy of the UAV after processing the packet, and movement after processing the packet. We use 70% of our data for training and the other 30% for testing. In our simulation study, we use accuracy, precision, and recall to measure in both contexts to determine the efficiency of the model, and also the amount of energy preserved is computed corresponding to every move. We also compare our approach with current state-of-the-art energy-preserving UAV movement control techniques which are compatible with the present application scenario. The proposed technique produced 6.5%, 34.5%, and 61.5% better results in terms of percentage of successful detection (PSD), composite energy consumption (CEC), and average delay (ADL), respectively.

Keywords: crop monitoring; energy efficient UAV; precision agriculture; SVM; trajectory optimization; UAV application; UAV-IoT



Citation: Al-Naeem, M.; Hafizur Rahman, M.M.; Banerjee, A.; Sufian, A. Support Vector Machine-Based Energy Efficient Management of UAV Locations for Aerial Monitoring of Crops over Large Agriculture Lands. *Sustainability* **2023**, *15*, 6421. <https://doi.org/10.3390/su15086421>

Academic Editors: Ali A. Alwan Al-juboori and Yonis Gulzar

Received: 25 February 2023

Revised: 1 April 2023

Accepted: 2 April 2023

Published: 10 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Precision agriculture has gained great importance these days, especially in countries with agriculture-based economies. Different tools and techniques can capture, process and analyze spatial and temporal data from various parts of large agricultural lands and combine those to extract complete information from it [1–6]. These changes brought about by technological development in agriculture are renamed the green movement [7–11]. As the demand for food commodities is expected to rise to 75% by 2050 [12], the involvement of technologies to deal with a series of different tasks related to agriculture is expected to become more relevant [13–15]. For this reason, in various countries UAVs are being increasingly applied in precision agriculture to safeguard crops in large cultivation lands [16–18]. The concept of the Internet of Things (IoT) has shown significant potential in the successful implementation of smart farming and can have more potential if it collaborates

with UAVs [19,20]. IoT devices catch signals from the respective lands on which they are deployed and these signals are sent to the UAV for further processing. After receiving these signals, the UAVs process those and take action accordingly. Earlier remote sensing was heavily used in cultivating fields where images were taken through satellite for various purposes such as monitoring vegetation status, irrigation, contamination detection of crops due to several diseases, and use of fertilizers [21,22].

In spite of all these applications, the process of remote sensing by capturing satellite images was not that efficient because of poor spatial resolution of images, huge costs due to frequently taking images of all the sites, and the complexity of analyzing each one of those with equal importance. However, the task can be simplified, and notifying farmers can be performed in a timely fashion in case of the presence of predatory insects, pests, etc., provided the tracking area is divided into smaller regions and the regions are classified as sensitive or insensitive based on history. Sensitive regions should be allotted more spatial and temporal resolution, that is, those regions should be revisited by the satellite more frequently and data should be captured with enough precision to detect even the small features. This method has been elaborated on and presented in our present article. SVM is used for classifying various regions of large agricultural lands as sensitive or insensitive from the history of earlier images processed by the UAV. Images are captured by a sensor and transmitted to the UAV whose cost is far lesser than satellite imagery. Therefore, our process is efficient in terms of both time and cost as supported by the simulation result where our proposed scheme SVM-UAV-ARMONT (SVM, UAV-based aerial monitoring) has been compared with state-of-the-art UAV-assisted aerial methods for monitoring agricultural lands based on the performance metrics—the percentage of successful detection (PSD), composite energy consumption (CEC), and average delay (ADL) with respect to a number of devices deployed. Results show that SVM-UAV-ARMONT saves a significant amount of energy and time while generating a huge successful detection percentage.

The rest of the article is organized as follows. Related work is discussed in Section 2; materials and methods are discussed in Section 3; history-based trajectory planning is mentioned in Section 4; simulation results and discussion are in Section 5; finally, a conclusion and the possible future scope are mentioned in Section 6.

2. Related Work

Research on drone and UAV technologies and their potential applications in the field of agriculture has started to be heavily explored in recent times [17,18,23–25]. Aerial vehicles are used to monitor the well-being of crops in fields, spraying of pesticides and various medicines to ensure the safety of the produced, etc. All these are being assisted by various technological features including machine learning, deep learning, and artificial intelligence [18,26,27]. Trajectory planning of UAVs has attracted the attention of researchers in many ways, both in single and multiple UAV systems, because with a selection of the most appropriate route a lot of energy can be saved [19,20,28–31]. Therefore, to make the process of real monitoring of cultivation fields very efficient, the cost has to be optimized and hence proper trajectory planning is extremely necessary for the operations of UAV systems. The concept of energy efficiency is implemented from perspectives:

- (i). Positions of changing stations [32–34].
- (ii). Energy-optimized execution of tasks requested by the sites [35–43].

The only advantage of trajectory planning based on the positions of changing stations is that energy efficiency is obtained only when UAVs are exhausted. On the other hand, task execution is a much more frequent phenomenon and if trajectories are planned based on task execution requests raised by different sites, then at each step some useful energy may be conserved. Therefore, trajectory planning depending upon task execution requests generates much better results (category (ii) mentioned above in terms of energy preservation compared to those in category (i) written above). Additionally, there is another point to be noted: if UAVs are exhausted less frequently due to the energy-efficient execution of

tasks, then the need for trajectory planning based on the positions of the changing stations is also reduced. Category (ii) can be further classified into three subcategories based on specialized techniques that are applied to plan flight paths of UAVs:

- (a). Ant-colony-based optimization [29–41].
- (b). Mixed-integer linear programming-based techniques [42,43].
- (c). Shortest-path-based planning [20,44].

An ant-colony optimization or ACO-based heuristic framework [75] concentrates on the optimal deployment of GCS for minimization of recharge cost of drones as well as UAVs. Mixed-integer linear programming (MILP) model-based trajectory planning is proposed in [42,43]. Here, sensor nodes are themselves embedded with the UAV capacity (namely SN-UAV) and in their context, a flat topology routing protocol is proposed for energy optimization. However, if the agricultural land is very large in size and a lot of sensor nodes have to be deployed at various portions of it, then embedding UAV capacity in each one of those will be difficult as well as expensive. Dijkstra's shortest-path-based algorithm is proposed where two-dimensional path planning is performed to avoid obstacles in the environment. Here, the weight of each edge is the inverse of the distance between the pair of sites, and the path with the highest weight is chosen by the UAV. This algorithm has many disadvantages. It does not consider anything other than the distance between the two sites. The UAV under this algorithm may traverse through all the regions which are unnecessary and there a huge amount of energy is wasted in spite of following the shortest paths for traversal.

Another possible categorization is:

- (i). UAV systems without facility of task delegation [34–43].
- (ii). UAV systems with facility of task delegation [33,44].

Task delegation contributes to overall system efficiency both in terms of energy and delay. Whenever a UAV feels exhausted and a nearby UAV is idle, the first UAV delegates some task to the second one. In that way, the first UAV is exhausted later, after completing more tasks, resulting in a smaller average delay and at the same time, has a better opportunity to fly closer to a charging station so that the energy required for recharging is decreased. Task delegation is extremely important for the performance effectiveness of multi-UAV systems. However, the algorithm is comparatively complex, which is its only disadvantage. Some multi-drone systems also appear in the literature where the task delegation method is used from one drone to another [33,45]. A single drone system in the context of a dock station is also used for this kind of purpose [46]. Hence, the trajectory of the drone is planned based on the heading angle and return points of agricultural UAVs are considered for the optimization of UAV paths.

Heading angle and return points of agricultural UAVs are considered for optimization of UAV paths in [47]. It applied techniques such as gravitational search and A* for path planning where it was shown that incidents such as non-spraying on fields have been reduced by 23.18%, whereas excess coverage of pesticide could be kept within the upper limit of 64.47%. However, for the detection of certain phenomena, some portions of a large agricultural land have to be considered to have greater importance, which has been ignored in this article [47].

The authors in [48] focus on the partial aspect that a UAV cannot actually position itself at least in the very beginning during the flight process. Therefore, the UAV applies a genetic algorithm as well as the A* algorithm (A*G) to plan its trajectory so that always the shortest route is followed to change position from one place to another.

A theoretical framework for cooperation between multiple UAV systems appears in [49], where a large agricultural land is divided into multiple regions with one UAV (MULT-UAV) dedicated to each of those regions. However, the method is very expensive because of the requirement to deploy such an average number of aerial vehicles. Flight path planning based on the artificial potential field method appears in [50,51], where the environment is considered, consisting of many obstacles and there are different threat

coefficients associated with different parts of an obstacle. The trajectory of UAVs is planned in such a manner that chances of collision are minimized.

3. Materials and Method

This section presents the materials and method of the proposed technique. First, a briefing of used abbreviations and symbols is presented in Section 3.1. The overall system architecture appears in Section 3.2, whereas functionality is described in detail in Section 3.3.

3.1. Abbreviations and Symbols

The paper used many abbreviations and symbols that are mentioned at the location where they are first used. Table 1 presents a brief of those abbreviations and symbols for better readability of the paper.

Table 1. Symbols, abbreviations, and their descriptions.

Symbol/Abbreviation	Brief Description
UAV	Unmanned Aerial Vehicle.
SVM	Support Vector Machine.
PSD	Percentage of Successful Detection.
CEC	Composite Energy Consumption.
ADL	Average Delay.
SVM-UAV-ARMONT	SVM and UAV-Based Aerial Monitoring.
GCS	Ground Charging Stations.
ACO	Ant Colony Optimization.
SN-UAV	Sensor Nodes of UAV.
A*GA	A*-Based Genetic Algorithm.
MULT-UAV	Multiple Regions with One UAV.
CH	Cluster Head.
HST	History Table.
SRF	Sensitive Regions First.
region-id	Denotes the Unique Identification Number of the Region.
region-area	Specifies its Area of the Specific Geographical Region.
Session-id	Unique Identification of the Session.
Num-of-sensors	The Number of Sensors Deployed in that Region.
detection-state	Acquires a Value of 1 if Something Unnatural is Detected in the Region.
Task-arrival	The Number of Tasks that are Currently Arriving in the Task Queue of the Current Node Per Unit of Time.
No-of-task-exec	The Number of Tasks that are Currently Executing from the Task Queue of the Current Node Per Unit of Time.
enrg_consumed	Energy Consumed.
res_enrg_UAV	Energy that Remained in UAV after Executing.
time_del	The Amount of Time Required by the UAV to Execute these Tasks.
Parameter C	Specifies the Penalty Parameter or the Error Term.
tas_que /tsk_queue	Task Queue.
length(task(j))	Indicates the Simulation of Lengths of all the Tasks in the Task Queue.
enr_unit_len_tsk(i)	The Amount of Energy that May be Required to Finish a Unit-Length Task.
del_unit_len_tsk(i)	The Amount of Time Delay Required for Executing a Unit-Length Task.
res_eng	Residual Energy.
AR	Region with Area.
W(Cap)	Weight of Cap.
AR	Region with Area.

3.2. System Architecture

The system architecture consists of a large irregularly shaped rectangular field that is divided into some regions based on soil quality, types of crops, etc. Different criteria with each region containing one or more sensor nodes as shown in Figure 1. One region comprises one or more non-intersecting geographical areas and sensor nodes are deployed in such a manner that each sensor node monitors one such geographical area within a region. These sensor nodes capture pictures of different regions at different interval times and send those to the head of their respective clusters to form batches of processing requests or tasks. There is only one UAV in the system, as shown in Figure 2, which flies over the cluster heads (CHs) and as soon as the UAV arrives over one particular CH, the CH sends the task of the task queue in batches; the UAV executes these tasks, that is, analyzes

all the images and takes action accordingly. This is initially performed as per Dijkstra's shortest path algorithm until and unless each CH is visited at least ten times and the UAV stores records about all these CHs and members. These records form a history and based on this history, certain regions are classified as sensitive while others are considered insensitive. Sensitive regions are subsequently visited more frequently by the UAV, that is, their temporal resolution increases, while insensitive regions are visited less frequently to improve the percentage of successful detection. Classification of regions is performed as sensitive or insensitive based on SVM.

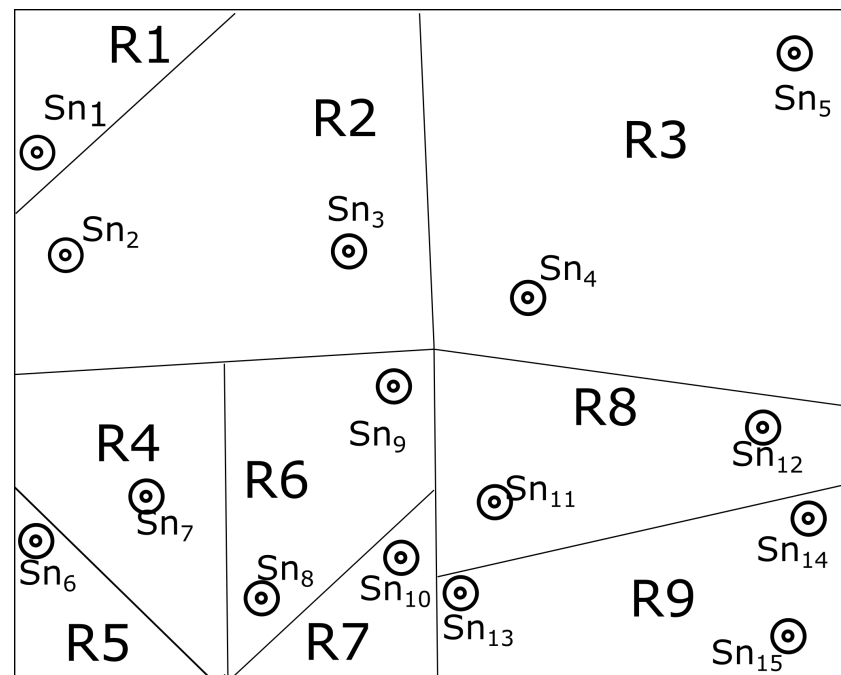


Figure 1. Sensors and regions of an agricultural field.

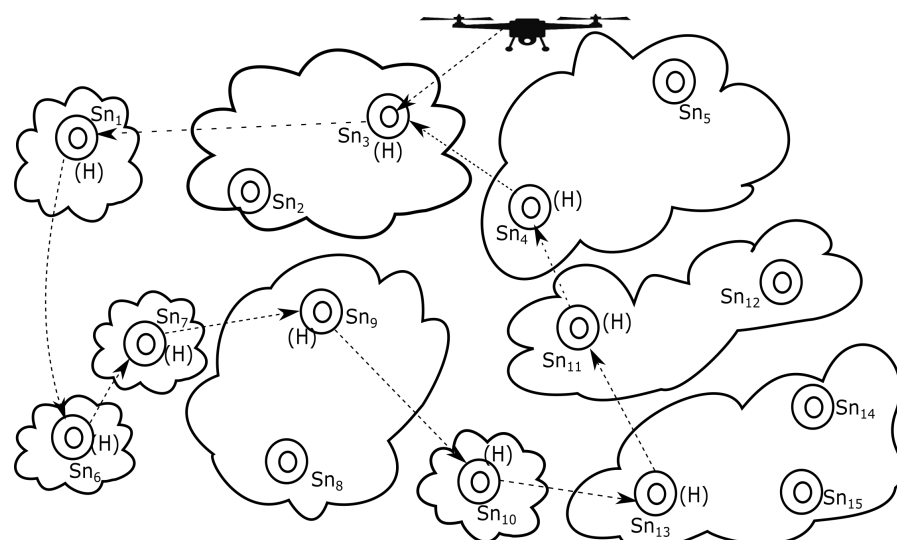


Figure 2. Typical cluster formation of the agricultural field.

3.3. SVM-Assisted Recognizing of Sensitive and Insensitive Regions

Whenever the UAV arrives on a particular CH for the first time, a history file (HST) consisting of at least ten records on each member of each CH shifts its trajectory planning strategy from Dijkstra's shortest path algorithm to the sensitive-regions-first (SRF) algorithm. Regions are classified as sensitive OR insensitive based on an SVM-based classifier

module that considers the attributes such as region-id, region-session-id, num-of-sensors, detection-stat area, task-arrival-rate, no-of-task-exec, eng-consumed, time-del, and res-energy-UAV. Here, region-id is the unique identification number of the region, region-area specifies its area of the specific geographical region, and session-id is a unique identification of the session in which the UAV flew over that particular region and executed tasks present in its task queue. num-of-sensors specifies the number of sensors deployed in that region, and detection-state acquires the value 1 if something unnatural is detected in the region, otherwise, it is set to 0. Task-arrival rate is the number of tasks that are currently arriving in the task queue of the current node per unit of time. No-of-task-exec is the number of tasks executed by the UAV in that session and the amount of energy consumed by the UAV to execute those tasks is given by `enrg_consumed` and energy that remained in the UAV after executing these is given by `res_enrg_UAV`. Time_del is the amount of time required by the UAV to execute these tasks.

Based on this entire history, all the regions are classified as sensitive or insensitive after executing the current set of tasks and inserting the information of the current session in the history table HST of the CH whose tasks are presently being executed by the UAV. Additionally, the new record is copied into the UAV in a variable named `new_rec_set` which is nothing but the set of new records. This new record set will be copied to the history table of the new CH after which the UAV will fly on the new CH.

In Section 3.2, we describe theoretical and practical implementations of SVM, whereas in Section 3.3 we describe trajectory planning after each CH is visited.

3.3.1. Support Vector Machine (SVM)

Support vector machine (SVM) is a supervised learning algorithm used for classification and regression tasks. It finds the best possible line or hyperplane that separates the different classes in the input data by maximizing the margin between the decision boundary and the nearest data points of each class. SVM can also handle non-linear classification problems by mapping the input data to a higher-dimensional feature space using a kernel function. SVM is effective even when the number of features is much larger than the number of training examples, and it is relatively insensitive to overfitting. However, SVM can be computationally expensive for large datasets and sensitive to the choice of kernel function and the regularization parameter [52].

The goal of the SVM technique is to create the best decision boundary that can segregate n -dimensional space into classes so that we can easily put the new data point in the correct class. This decision boundary is called a hyperplane. SVM chooses the utmost vectors (points) that are used in creating the hyperplane. These utmost cases are called support vectors, and hence the algorithm is termed the support vector machine.

3.3.2. Practical Implementation of the SVM

SVM can be useful in precision agriculture for tasks such as crop classification, disease detection, and yield prediction. SVM can handle high-dimensional and non-linear data, which are common in precision agriculture applications. SVM can also provide accurate and reliable predictions, making it useful for decision-making in precision agriculture. Additionally, SVM is relatively insensitive to overfitting, which can be helpful when working with limited or noisy data. This is the primary reason to utilize SVM for this problem.

The history table is stored in a CSV file. The percentages of training and testing data of the HST table of all CHs are 70% and 30%, respectively. The values of accuracy, precision, and recall for training and testing data are 93.25%, 91.71%, 92.54%, 91.67%, 90.11%, and 90.9%, respectively. All of these values are greater than 80%, which means that construct of the model is fit enough to explore the relationships between sensitive decisions and their factors in the HST table. The parameters of the SVM are shown below in Table 2.

Table 2. Names and values of various simulation parameters of SVM.

Name of the Parameter	Value
C	100.00
class_weight	None
HST_tablesize	500
Cache_size	500
coef0	0.0
decision_function_shape	ovr
degree	3
gamma	auto_deprecated
kernel	rbf
max_iter	−1
probability	False
random_state	None
Shrinking	True
tol	0.001
ver_bose	False

Parameter C specifies the penalty parameter or the error term. It is nothing but the degree of correct classification and its minimum value is 2 since it is a two-class problem. However, for values higher than this, “there is no way the SVM classifier can miss classify any single point”. *class_weight* is the next parameter that depends on possible labels. Here, the labels are 0 to 1 for insensitive and sensitive regions, respectively. This parameter is set to none so that both classes get equal weight. The size of the history of CH is limited to 500. Whenever a new record is inserted, an older record has to be deleted if the no. of records exceeded 500. From [53–55], values of *c* higher than two for arbitrary classification problem generates an accuracy of more than 90%. The minimum required sample size is 400 to achieve an accuracy close to 93%. Based on this, we have decided the value of C to be 100 and *HST_tablesize* to be 500. *coef0* is an independent term that is applicable only in the case of poly and sigmoid types of functions. Therefore, in the present scenario this is set to 0.0, “ovr” or *one_verses_rest* is set to the shape of decision function, *one_verses_rest* classifier grabs a class and creates a binary label for whenever a data point belongs to that class or not. The values of degree and gamma are 3 and auto deprecated, respectively. ‘degree’ is nothing but the degree of the polynomial and its default value is 3. Gamma is set to auto_deprecated meaning that no specific value of it has been passed. The kernel cache size is set to 500. Keeping parity with the table size of his history or *HST_tablesize*, rbf is the default kernel type, *max_iter* specifies the upper limit on iteration, which, if set to −1 specifies no limit on the number of iterations. A probability option is required to compute the probability estimates. By default, it is set to false. The random state is ignored when the probability is false. Similarly, the default value for the shrinking heuristic is true. *tol* specifies the tolerance value for the stopping criterion and we go for a default value of 0.001 for this parameter as well. *verbose* is a boolean variable that is set to false by default because, otherwise, it may not work in a multi-threaded environment. Default values have been taken from literature on SVM [53–55].

4. History-Based Trajectory Planning

The proposed technique SVM_UAV_ARMONT estimates the trajectory based on previous activities. As already mentioned in Section 3.2, attributes of the history table of the UAV are:

- (i) *region_id*;
- (ii) *region_area*;
- (iii) *session_id*;
- (iv) *num_of_sensors*;
- (v) *detection_stat*;

- (vi) tot_len_of_task_exec (it specifies the total length of tasks that are executed in that session);
- (vii) no_of_task_exec;
- (viii) enrg_consumed;
- (ix) time_del;
- (x) res_enrg_UAV.

Suppose the UAV arrives over the CH C_i , with residual energy $res_en(i)$. The region_id is nothing but i , which is a unique identification of the region. Assume that C_i monitors a region with area $AR(i)$ where the area of the large agricultural field is F . Up to now there was information about v number of sessions associated with C_i , so the current session in $(v+1)$ th session. num_of_sessions for the particular CH remains the same across all sessions. detection_stat in $(v+1)$ th station is denoted by SVM. Where ARMA_prediction is the function that predicts task arrival rate in $(v+1)$ th session based on $tar(1), tar(2), \dots, tar(v)$. The number of tasks to be executed in the current session is the number of tasks present in task queue of C_i . It is readily available in the current session and does not have to be computed.

Total energy $enrg_i(v+1)$ to be consumed in the current session is modeled as the average of $enrg_i^1(v+1)$ and $enrg_i^2(v+1)$ as shown in (1).

$$enrg_i(v+1) = \begin{cases} \frac{1}{2}[enrg_i^1(v+1)+enrg_i^2(v+1)] & \text{if } enrg_i^2(v+1) > enrg_i^1(v+1) \\ enrg_i^1(v+1) & \text{otherwise} \end{cases} \quad (1)$$

where $enrg_i^1(v+1)$ is the straightforward computation of energy from composite task length currently present in the task queue of C_i , and $enrg_i^2(v+1)$ is ARMA model predicted consumption of energy, where prediction is performed based on actual energy consumed in previous sessions such as $en_rg_i(1), en_rg_i(2), en_rg_i(3), \dots, en_rg_i(v)$.

$$enrg_i^1(v+1) = enr_unit_len_task(i) \times \sum_{j=1}^{|tsk_queue(i)|} length(task_i(j)) \quad (2)$$

where $task_que(i)$ is set of tasks in task queue of C_i and, therefore, $|tsk_queue(i)|$ is number of tasks in task queue of C_i . So, $task_i(1)$ is the first task in task queue $tsk_que(i)$ of C_i . Similarly, $task_i(2)$ is the second task of $tsk_que(i)$ of C_i and so on. In general, we can say that $task_i(j)$ is the j -th task in the task queue of C_i where $i \leq j \leq \log|tas_que(i)|$. $length(task_i(j))$ specifies the length of the task denoted by the j -th task in the task queue of c_i . So $\sum_{j=1}^{|tsk_queue(i)|} length(task_i(j))$ indicates the simulation of lengths of all the tasks in the task queue of C_i . $enr_unit_len_tsk(i)$ is the amount of energy that may be required to finish a unit-length task by C_i . So, $enrg_i(v+1)$ is the total energy that will be required to complete all tasks in the task queue of C_i .

The formulation of $enrg_i^2(v+1)$ appears in (3) and it is based on the ARMA model,

$$enrg_i^2(v+1) = ARMA_prediction(en_rg_i(1), en_rg_i(2), \dots, en_rg_i(v)) \quad (3)$$

Similar to energy, time delay $time_del_i(v)$ is modeled in (4) as the average of $time_del_i^1(v+1)$ and $time_del_i^2(v+1)$ where $time_del_i^1(v+1)$ specifies exact time delay required to complete all the tasks present in task queue C_i where $time_del_i^2(v+1)$ is the time delay predicted by the ARMA model from history of time delay of previous sessions in the HST table.

$$time_del_i(v+1) = \begin{cases} \frac{1}{2}[time_del_i^1(v+1)+time_del_i^2(v+1)] & \text{if } time_del_i^1(v+1) > time_del_i^2(v+1) \\ time_del_i^1(v+1) & \text{otherwise} \end{cases} \quad (4)$$

where

$$time_del_i^1(v+1) = del_unit_len_task(i) \times \sum_{j=1}^{|task_queue(i)|} length(task_i(j)) \text{ and}$$

$$time_del_i^2(v+1) = ARMA_prediction(del_rg_i(1), del_rg_i(2), \dots, del_rg_i(v))$$

Here, $del_unit_len_task(i)$ is the amount of time delay required for executing a unit-length task. $del_rg_i(1), del_rg_i(2), \dots, del_rg_i(v)$ are time delays specified in v earlier sessions as per the history of task execution of C_i .

ARMA model-based prediction is incorporated here because the situation that all tasks are successfully computed in one go is nothing but the ideal situation which does not always happen. Certain tasks have to be redone more than once. So, in order to estimate the energy and time delay that may be required for the execution of all those tasks in the task queue of C_i , including the non-ideal situations, we apply the ARMA model for both the energy and delay contexts.

After executing all the tasks in the task queue of C_i , residual energy $res_eng_i(UAV)$ of the UAV will be formulated as (5).

$$res_eng_i(UAV) = res_eng_{i-1}(UAV) - enrg_i(v+1) \quad (5)$$

From the history table (HST) of C_i , certain regions are identified as sensitive while others are identified as insensitive. Assume that after C_i set of sensitive regions remaining to be traversed in the current run is given by $CLS_SET_i(UAV)$, such that, $CLS_SET_i(UAV) = \{C_{a1}, C_{a2}, \dots, C_{au}\}$ for trajectory planning, we need to determine which CH C_{ap} should be visited after C_i , where $1 \leq p \leq u$, s.t., u is the number of CHs of the sensitive region in $CLS_SET_i(UAV)$. The weight of C_{ap} is given by $W(C_{ap})$ and formulated in (6). It is nothing but a function of estimated energy consumption required for executing tasks in C_{ap} as per the HST table of C_i . Energy consumption required for traveling from C_i to C_{ap} , estimated time required for execution of task in C_{ap} as per HST table of C_i , area covered by C_{ap} with respect to the area of the entire agricultural field. The number of earlier sessions in which C_{ap} was found sensitive as per the HST table of C_i with respect to the total number of earlier sessions involving C_{ap} .

$$W(C_{ap}) = [f_i^1(C_{ap}) \times f_i^2(C_{ap}) \times f_i^3(C_{ap})]^{-1} \times (f_i^4(C_{ap}) \times f_i^5(C_{ap}) \times f_i^6(C_{ap})) \quad (6)$$

$$f_i^1(C_{ap}) = k' \left\{ (x_i - x_{ap})^2 + (y_i - y_{ap})^2 \right\} \quad (7)$$

$$f_i^2(C_{ap}) = \begin{cases} \frac{1}{2} [enrg_{ap}^1(v+1) + enrg_{ap}^2(v+1)], & \text{if } enrg_{ap}^2(v+1) > enrg_{ap}^1(v+1) \\ enrg_{ap}^1(v+1), & \text{Otherwise} \end{cases} \quad (8)$$

$$enrg_{ap}^1(v+1) = enr_unit_len_tsk(ap) \times (ARMA_prediction(task_len_{ap}(1), task_len_{ap}(2), \dots, task_len_{ap}(v))) \quad (9)$$

$enrg_{ap}^2(v+1)$ has already been formulated in (3).

$$f_i^3(C_{ap}) = \begin{cases} \frac{1}{2} [time_del_{ap}^1(v+1) + time_del_{ap}^2(v+1)], & \text{if } time_del_{ap}^2(v+1) > time_del_{ap}^1(v+1) \\ time_del_{ap}^2(v+1), & \text{Otherwise} \end{cases} \quad (10)$$

$$time_del_{ap}^1(v+1) = del_unit_len_tsk(ap) \times (ARMA_prediction(task_len_{ap}(1), task_len_{ap}(2), \dots, task_len_{ap}(v))) \quad (11)$$

$$f_i^4(ap) = \frac{area(ap)}{F} \quad (12)$$

$$f_i^5(ap) = \frac{v'(ap)}{v(ap)} \quad (13)$$

$v'(ap)$ is the number of sessions in which ap was detected as sensitive, and $v(ap)$ is the number of sessions for which data involving ap is stored in the HST table of C_i .

$$f_i^6(C_{ap}) = \frac{enrg_{ap}^1(v+1)}{enr_unit_len_tsk} \quad (14)$$

In this way, the weight values of all CHs are calculated for all p , s.t., $1 \leq qp \leq u$. In each pass, the CH with height weight is chosen as the next destination of the UAVs after traversing the sensitive region q times, insensitive regions are visited once, and the value of q is based on the ratio of areas covered by sensitive and insensitive regions. Its value is computed in (15).

$$q = \frac{\sum_{p=1}^u area(ap)}{F - \sum_{p=1}^u area(ap)} \quad (15)$$

For traversing insensitive regions, weight values of CH are also calculated and the CH with the next weight is traversed next. A flow chart of SVM_UAV_ARMONT before and during the classification of regions is shown in Figure 3.

When the system begins operation, initially the agricultural field is divided into some regions, sensors are deployed, and then CHs are elected. Turn = 0 specifies that traversing of CHs has just begun and the first CH is about to be traversed. The first CH is identified by arranging CHs using Dijkstra's shortest path algorithm. For all subsequent traversals, the turn will remain 1. After the first clustered is visited by the UAV, the turn will be set to 1. The record will contain all information (region_area, session_id, num_of_sessions, tot_len_of_task_exec, no_of_task_exec, enrg_consumed, time_del, res_enrg_UAV) about current CH and the same information will be stored in history table or HST of the current CH. Additionally, the current CH will be stored in a preview or previous CH record which is supposed to be the next CH. This will continue until the HST table contains ($N \times 10$) records. Then, classification is performed where a set of sensitive regions will be denoted as α and a set of insensitive regions will be denoted as β ; q denotes the ratio of the area of sensitive and insensitive regions. Sensitive regions will be traversed q number of times whereas insensitive regions will be traversed once. The technique is similar to the previous one, only the next destination CH will be chosen based on weight values computed using the SVM_UAV_ARMONT technique. After classifying regions as sensitive or insensitive, re-classification takes place after a time interval ζ . If it is found that the outcome of region labels (sensitive or insensitive) remains the same then ζ is increased to $2 \times \zeta$, that is, doubled. On the other hand, if the label of at least one region changes, then the re-classification interval is halved, that is, becomes $\zeta/2$. Accordingly, the route of the UAV also changes. Therefore, classification is not always permanent and it depends on behavior regions. A flowchart of the overall system appears in Figure 4.

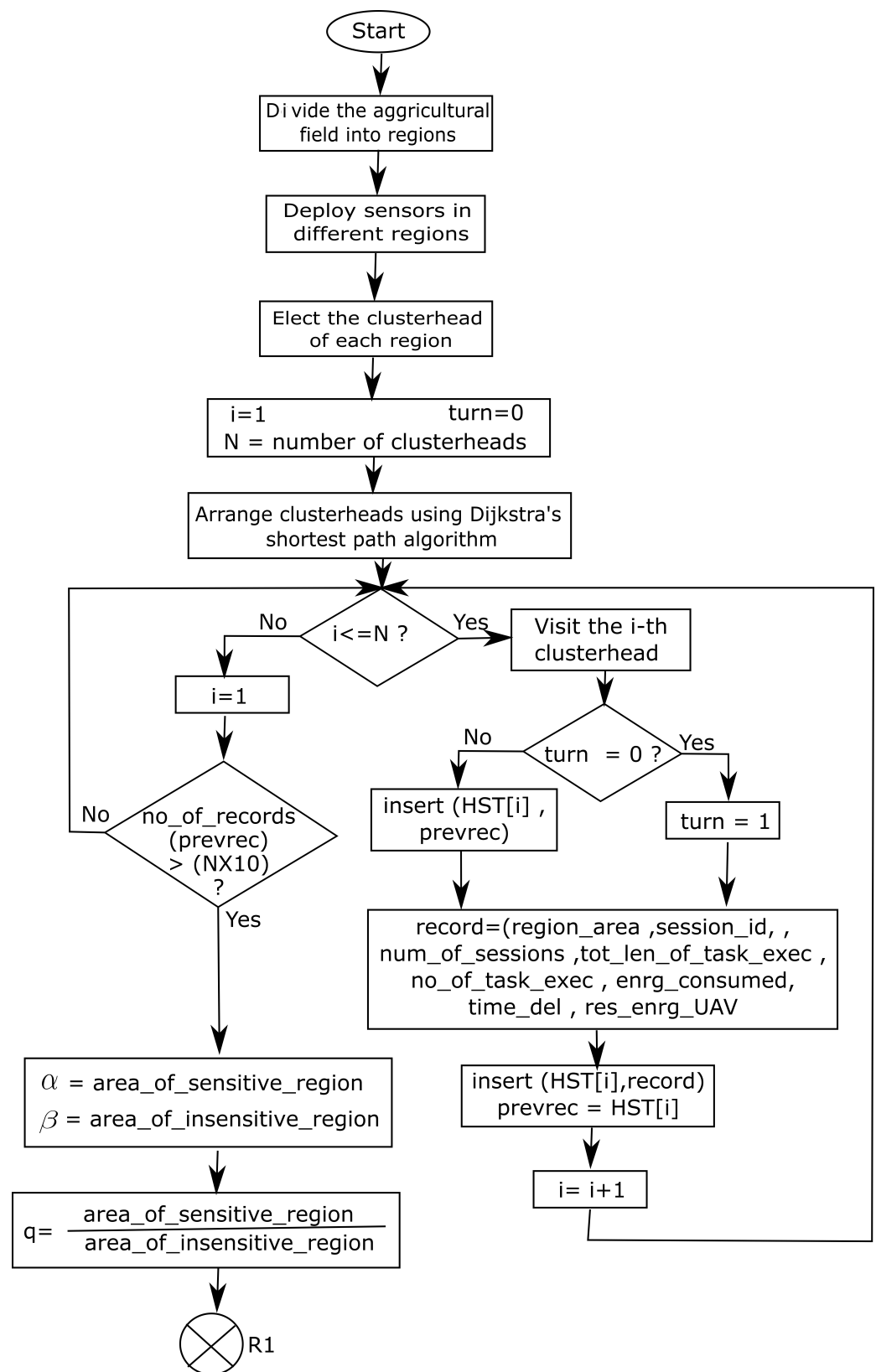


Figure 3. Flowchart of SVM_UAV_ARMONT before and during the classification of regions.

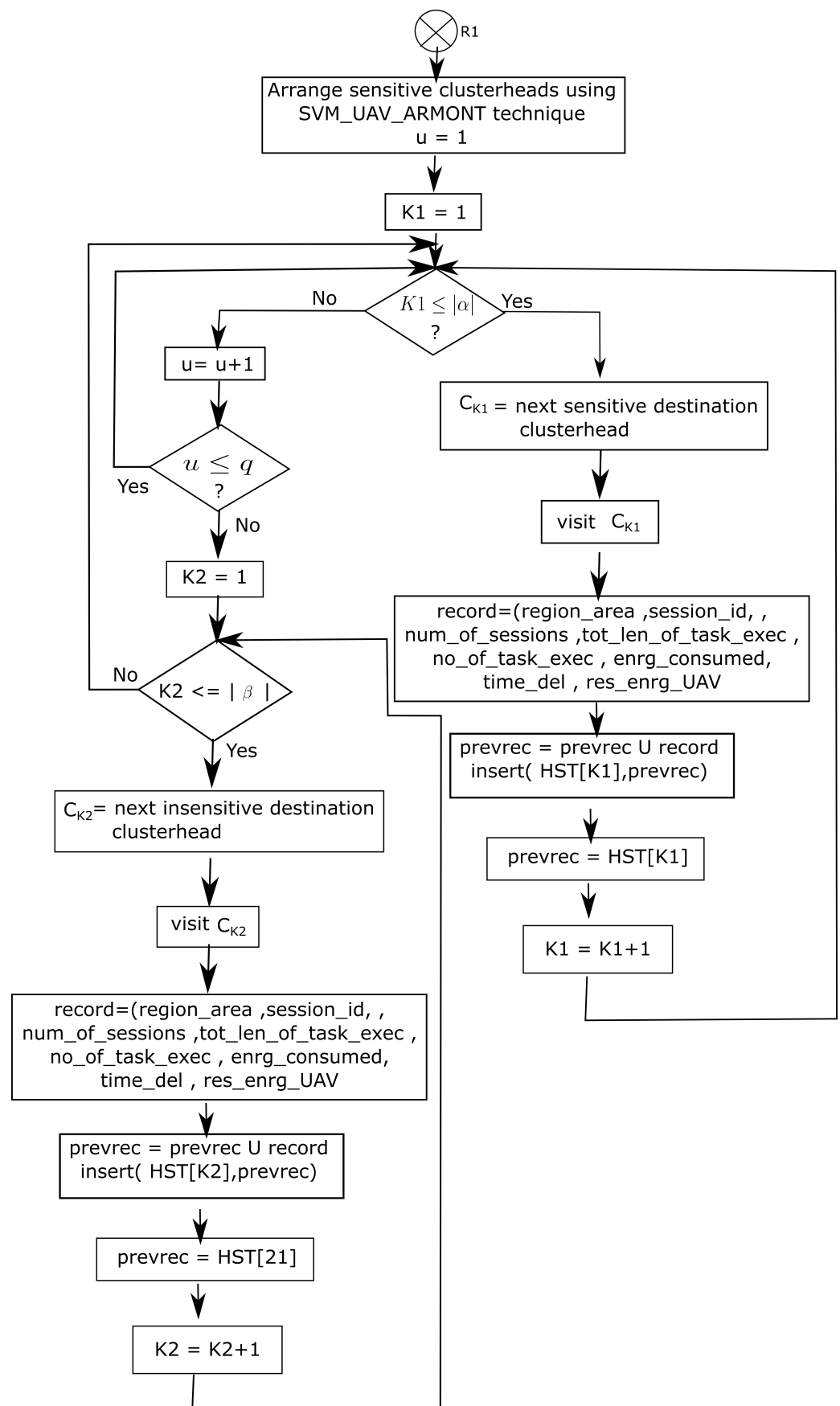


Figure 4. Flowchart of the overall system after the classification of regions.

5. Simulation Results and Discussion

The proposed SVM_UAV_ARMONT is tested in a simulation setup and the results are presented in this section.

Simulation Setup

Table 3 presents the summarized values of all simulation parameters setup that are used in this experiment.

Table 3. Names and values of various simulation parameters.

Name of the Parameter	Value
Coverage area	$300 \times 300 \text{ m}^2$
No. of regions	10
Min no. of devices in a cluster	3
Max no. of devices in a cluster	20
Number of IoT devices	100, 300, 500, 700, 900
Number of UAVs	1, 3, 5, 7, 9
Hovering altitude of UAV	6 m
No. of CPU cycles/s in an UAV	0.5G cycles/s
UAV speed	20 m/s
Transmission bandwidth	150 khz
Channel gain	−20 dB
Noise power	−50 dBm
Initial value of re-classification interval	1 h

In SN_UAV, all sensor nodes have UAV capability embedded. However, when the agricultural field is large and a lot of sensors need to be deployed, then the associated cost will be very high. Additionally, there is no application of the concept of delayed processing of batches. Therefore, for each abnormality detected in the field, the corresponding sensor-cum-UAV has to process the task individually and fly. During the time it flies, some of the events will miss detection, leading to decreased PSD, especially when the number of sensors is very high, i.e., the agricultural field monitor is very large in size, for which a huge number of sensors have to be deployed. Moreover, since the task of UAV is performed, energy consumption is much higher. A8_GA has to be performed as well at ground stations and the Huge_no_of estimated task means more abnormalities detected.

During this research, sensor-cum-UAV nodes cannot sense the environment and miss some of the detection. As far as A*-GA is concerned, a genetic algorithm is used for planning the UAV trajectory when only energy consumption is considered in the fitness function. The next destination A*-GA is the node to go to which minimum energy is consumed by the UAV. However, it does not differentiate between the areas where most of the abnormalities have been detected in the recent past (sensitive regions) and the areas where no abnormality has been detected recently (insensitive regions). UAVs should spend most of the time in sensitive regions, which is performed in SVM_UAV_ARMONT but not its state-of-the-art competitors SN-UAV and A*-GA. Giving equal importance to both sensitive and insensitive regions is a waste of both energy and time. With increased exhaustion of the UAV, it will have to be recharged frequently and during recharge, it will expectedly miss detection of some abnormalities, as shown in Figure 5. Additionally, the energy consumption of SN-UAV and A8-GA will be much higher than SVM_UAV_ARMONT as shown in Figure 6. Along with that, the time efficiency of our proposed scheme is graphically presented in Figure 7. SVM_UAV_ARMONT chooses the next CH destination based on energy consumption,

delay, amount of work that might have to be done in the new destination, and the amount of area covered by it. The destination that covers a huge area with a huge estimated number of tasks with less energy and delay is preferred the most. The huge number of the estimated task means more abnormalities are detected. In brief, the proposed technique produced 6.5%, 34.5%, and 61.5% better results in terms of percentage of successful detection (PSD), composite energy consumption (CEC), and average delay (ADL), respectively, over the state of the art.

Another reason behind the performance enhancement of our proposed scheme is the adaptive adjustment of the re-classification interval. After performing re-classification for the first time, if there is a change in the sensitive/insensitive label of at least one region, then the re-classification interval is halved, otherwise, it is doubled. On the other hand, for mentioned competitors of our proposed technique, the re-classification interval is fixed, hence, proper detection of sensitivity is lower in those. For example, suppose in one particular region the disease of crops has been cured and it need not be considered as sensitive anymore. So, these real-time cases will be more accurately identified by over proposed technique.

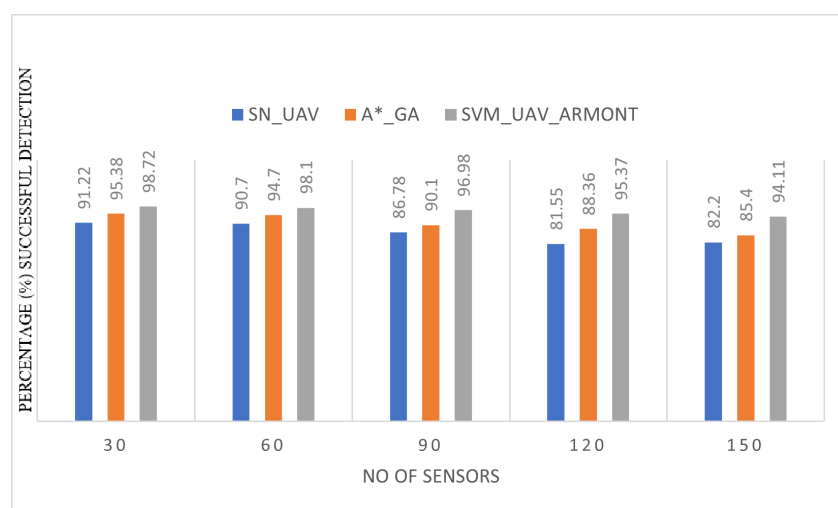


Figure 5. Percentage of successful detection (PSD) with respect to different nos. of sensors.

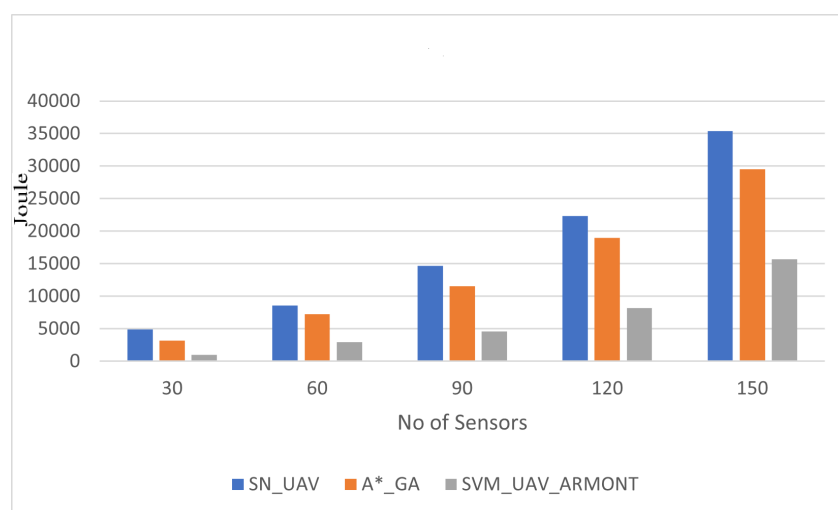


Figure 6. Composite energy consumption (CEC) with respect to the no. of sensors.

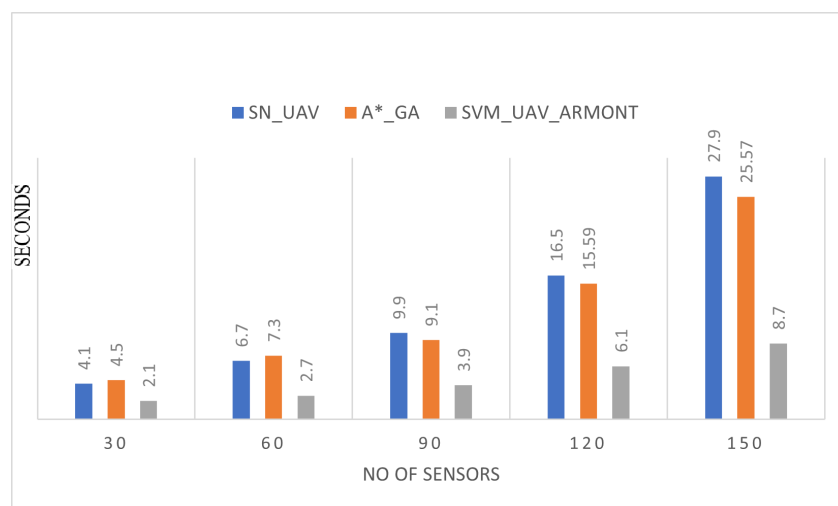


Figure 7. Average delay (ADL) with respect to the no. of sensors.

6. Conclusions and Future Scope

The present article proposed a single-UAV system that executes the tasks of IoT devices deployed in various regions with large agricultural lands. Both energy consumption and time delay have been considered in this respect although energy has been emphasized because our goal was to develop an energy-efficient system for aerial monitoring of crops. The main reason behind the novelty and performance enhancement of SVM_UAV_ARMONT is the differentiation between sensitive and insensitive regions. Traversing more over sensitive regions yields a higher percentage of detection of abnormalities, whereas avoiding unnecessary traversals over comparatively insensitive regions results in a lot of energy saving improving the efficiency of the system. SVM is needed to classify each region as sensitive or insensitive.

This system is a single-UAV system and we have a plan to expand it to a multi-UAV system, which will have additional issues to resolve such as trajectory planning avoiding collisions among multiple UAVs, task delegations from busier to less busy ones, and so on.

Author Contributions: All authors are equally contributed to this paper. All authors have read and agreed to this version of the manuscript.

Funding: This work was supported through the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia through project number INST023. The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia, for funding this research work (project number INST023).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Zhang, N.; Wang, M.; Wang, N. Precision agriculture—A worldwide overview. *Comput. Electron. Agric.* **2002**, *36*, 113–132. [CrossRef]
2. Gebbers, R.; Adamchuk, V.I. Precision agriculture and food security. *Science* **2010**, *327*, 828–831. [CrossRef] [PubMed]
3. Pierpaoli, E.; Carli, G.; Pignatti, E.; Canavari, M. Drivers of precision agriculture technologies adoption: A literature review. *Procedia Technol.* **2013**, *8*, 61–69. [CrossRef]
4. Shafi, U.; Mumtaz, R.; García-Nieto, J.; Hassan, S.A.; Zaidi, S.A.R.; Iqbal, N. Precision agriculture techniques and practices: From considerations to applications. *Sensors* **2019**, *19*, 3796. [CrossRef]
5. Couliably, S.; Kamsu-Foguem, B.; Kamissoko, D.; Traore, D. Deep learning for precision agriculture: A bibliometric analysis. *Intell. Syst. Appl.* **2022**, *16*, 200102. [CrossRef]

6. Gulzar, Y. Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique. *Sustainability* **2023**, *15*, 1906. [\[CrossRef\]](#)
7. Jez, J.M.; Lee, S.G.; Sherp, A.M. The next green movement: Plant biology for the environment and sustainability. *Science* **2016**, *353*, 1241–1244. [\[CrossRef\]](#)
8. Oliver, M.A.; Bishop, T.F.; Marchant, B.P. (Eds.) *Precision Agriculture for Sustainability and Environmental Protection*; Routledge Abingdon: Oxfordshire, UK, 2013.
9. Duhan, J.S.; Kumar, R.; Kumar, N.; Kaur, P.; Nehra, K.; Duhan, S. Nanotechnology: The new perspective in precision agriculture. *Biotechnol. Rep.* **2017**, *15*, 11–23. [\[CrossRef\]](#)
10. Lindblom, J.; Lundström, C.; Ljung, M.; Jonsson, A. Promoting sustainable intensification in precision agriculture: Review of decision support systems development and strategies. *Precis. Agric.* **2017**, *18*, 309–331. [\[CrossRef\]](#)
11. Zhang, P.; Guo, Z.; Ullah, S.; Melagraki, G.; Afantitis, A.; Lynch, I. Nanotechnology and artificial intelligence to enable sustainable and precision agriculture. *Nat. Plants* **2021**, *7*, 864–876. [\[CrossRef\]](#)
12. Baldos, U.L.C.; Hertel, T.W. Global food security in 2050: The role of agricultural productivity and climate change. *Aust. J. Agric. Resour. Econ.* **2014**, *58*, 554–570. [\[CrossRef\]](#)
13. Rosegrant, M.W.; Koo, J.; Cenacchi, N.; Ringler, C.; Robertson, R.D.; Fisher, M.; Cox, C.M.; Garrett, K.; Perez, N.D.; Sabbagh, P. *Food Security in a World of Natural Resource Scarcity: The Role of Agricultural Technologies*; The International Food Policy Research Institute (IFPRI); Washington, DC, USA, 2014.
14. Bhakta, I.; Phadikar, S.; Majumder, K. State-of-the-art technologies in precision agriculture: A systematic review. *J. Sci. Food Agric.* **2019**, *99*, 4878–4888. [\[CrossRef\]](#)
15. Mamat, N.; Othman, M.F.; Abdulghafor, R.; Alwan, A.A.; Gulzar, Y. Enhancing Image Annotation Technique of Fruit Classification Using a Deep Learning Approach. *Sustainability* **2023**, *15*, 901. [\[CrossRef\]](#)
16. Radoglou-Grammatikis, P.; Sarigiannidis, P.; Lagkas, T.; Moscholios, I. A compilation of UAV applications for precision agriculture. *Comput. Netw.* **2020**, *172*, 107148. [\[CrossRef\]](#)
17. Rejeb, A.; Abdollahi, A.; Rejeb, K.; Treiblmaier, H. Drones in agriculture: A review and bibliometric analysis. *Comput. Electron. Agric.* **2022**, *198*, 107017. [\[CrossRef\]](#)
18. Hafeez, A.; Husain, M.A.; Singh, S.; Chauhan, A.; Khan, M.T.; Kumar, N.; Chauhan, A.; Soni, S. Implementation of drone technology for farm monitoring & pesticide spraying: A review. *Inf. Process. Agric.* **2022**. [\[CrossRef\]](#)
19. Banerjee, A.; Sufian, A.; Paul, K.K.; Gupta, S.K. Edtp: Energy and delay optimized trajectory planning for uav-iot environment. *Comput. Netw.* **2022**, *202*, 108623. [\[CrossRef\]](#)
20. Rahman, M.M.H.; Al-Naeem, M.; Banerjee, A.; Sufian, A. EETO-GA: Energy Efficient Trajectory Optimization of UAV-IoT Collaborative System Using Genetic Algorithm. *Appl. Sci.* **2023**, *13*, 2535. [\[CrossRef\]](#)
21. Sishodia, R.P.; Ray, R.L.; Singh, S.K. Applications of remote sensing in precision agriculture: A review. *Remote. Sens.* **2020**, *12*, 3136. [\[CrossRef\]](#)
22. Dakir, A.; Barramou, F.; Alami, O.B. Opportunities for artificial intelligence in precision agriculture using satellite remote sensing. In *Geospatial Intelligence: Applications and Future Trends*; Springer: Cham, Switzerland, 2022; pp. 107–117.
23. Velusamy, P.; Rajendran, S.; Mahendran, R.K.; Naseer, S.; Shafiq, M.; Choi, J.G. Unmanned Aerial Vehicles (UAV) in precision agriculture: Applications and challenges. *Energies* **2021**, *15*, 217. [\[CrossRef\]](#)
24. Maddikunta, P.K.R.; Hakak, S.; Alazab, M.; Bhattacharya, S.; Gadekallu, T.R.; Khan, W.Z.; Pham, Q.V. Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges. *IEEE Sens. J.* **2021**, *21*, 17608–17619. [\[CrossRef\]](#)
25. Aslan, M.F.; Durdu, A.; Sabanci, K.; Ropelewska, E.; Gültekin, S.S. A comprehensive survey of the recent studies with UAV for precision agriculture in open fields and greenhouses. *Appl. Sci.* **2022**, *12*, 1047. [\[CrossRef\]](#)
26. Chen, C.J.; Huang, Y.Y.; Li, Y.S.; Chen, Y.C.; Chang, C.Y.; Huang, Y.M. Identification of fruit tree pests with deep learning on embedded drone to achieve accurate pesticide spraying. *IEEE Access* **2021**, *9*, 21986–21997. [\[CrossRef\]](#)
27. Khan, S.; Tufail, M.; Khan, M.T.; Khan, Z.A.; Iqbal, J.; Wasim, A. Real-time recognition of spraying area for UAV sprayers using a deep learning approach. *PLoS ONE* **2021**, *16*, e0249436. [\[CrossRef\]](#) [\[PubMed\]](#)
28. Liu, J.; Wang, X.; Bai, B.; Dai, H. Age-optimal trajectory planning for UAV-assisted data collection. In Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, USA, 15–19 April 2018; pp. 553–558.
29. Sun, C.; Ni, W.; Wang, X. Joint computation offloading and trajectory planning for UAV-assisted edge computing. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 5343–5358. [\[CrossRef\]](#)
30. Zhu, B.; Bedeer, E.; Nguyen, H.H.; Barton, R.; Henry, J. UAV trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9540–9554. [\[CrossRef\]](#)
31. Ivić, S.; Crnković, B.; Grbčić, L.; Matleković, L. Multi-UAV trajectory planning for 3D visual inspection of complex structures. *Autom. Constr.* **2023**, *147*, 104709. [\[CrossRef\]](#)
32. Basiri, A.; Mariani, V.; Silano, G.; Aatif, M.; Iannelli, L.; Glielmo, L. A survey on the application of path-planning algorithms for multi-rotor UAVs in precision agriculture. *J. Navig.* **2022**, *75*, 364–383. [\[CrossRef\]](#)
33. Mukhamediev, R.I.; Yakunin, K.; Aubakirov, M.; Assanov, I.; Kuchin, Y.; Symagulov, A.; Levashenko, V.; Zaitseva, E.; Sokolov, D.; Amirgaliyev, Y. Coverage path planning optimization of heterogeneous UAVs group for precision agriculture. *IEEE Access* **2023**, *11*, 5789–5803. [\[CrossRef\]](#)

34. Vazquez-Carmona, E.V.; Vasquez-Gomez, J.I.; Herrera-Lozada, J.C.; Antonio-Cruz, M. Coverage path planning for spraying drones. *Comput. Ind. Eng.* **2022**, *168*, 108125. [\[CrossRef\]](#)
35. Mohsan, S.A.H.; Othman, N.Q.H.; Khan, M.A.; Amjad, H.; Żywiołek, J. A comprehensive review of micro UAV charging techniques. *Micromachines* **2022**, *13*, 977. [\[CrossRef\]](#) [\[PubMed\]](#)
36. Wang, Y.; Su, Z.; Zhang, N.; Li, R. Mobile wireless rechargeable UAV networks: Challenges and solutions. *IEEE Commun. Mag.* **2022**, *60*, 33–39. [\[CrossRef\]](#)
37. Hoseini, S.A.; Hassan, J.; Bokani, A.; Kanhere, S.S. In situ mimo-wpt recharging of uavs using intelligent flying energy sources. *Drones* **2021**, *5*, 89. [\[CrossRef\]](#)
38. Angurala, M.; Bala, M.; Bamber, S.S. Wireless battery recharging through UAV in wireless sensor networks. *Egypt. Inform. J.* **2022**, *23*, 21–31. [\[CrossRef\]](#)
39. Li, J.; Xiong, Y.; She, J. An improved ant colony optimization for path planning with multiple UAVs. In Proceedings of the 2021 IEEE International Conference on Mechatronics (ICM), Kashiwa, Japan, 7–9 March 2021; pp. 1–5.
40. Gao, Z.; Zhu, J.; Huang, H.; Yang, Y.; Tan, X. Ant colony optimization for UAV-based intelligent pesticide irrigation system. In Proceedings of the 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Dalian, China, 5–7 May 2021; pp. 720–726.
41. Shafiq, M.; Ali, Z.A.; Israr, A.; Alkhamash, E.H.; Hadjouni, M.; Jussila, J.J. Convergence analysis of path planning of multi-UAVs using max-min ant colony optimization approach. *Sensors* **2022**, *22*, 5395. [\[CrossRef\]](#)
42. Zhang, L.; Zhou, Z.; Zhang, F.M. Mixed integer linear programming for UAV trajectory planning problem. *Appl. Mech. Mater.* **2014**, *541*, 1473–1477. [\[CrossRef\]](#)
43. Savin, S.; Al Badr, A.; Devitt, D.; Fedorenko, R.; Klimchik, A. Mixed-Integer-Based Path and Morphing Planning for a Tensegrity Drone. *Appl. Sci.* **2022**, *12*, 5588. [\[CrossRef\]](#)
44. Samir, M.; Sharafeddine, S.; Assi, C.M.; Nguyen, T.M.; Ghayeb, A. UAV trajectory planning for data collection from time-constrained IoT devices. *IEEE Trans. Wirel. Commun.* **2019**, *19*, 34–46. [\[CrossRef\]](#)
45. Tevyashov, G.K.; Mamchenko, M.V.; Migachev, A.N.; Galin, R.R.; Kulagin, K.A.; Trefilov, P.M.; Onisimov, R.O.; Goloburkin, N.V. Algorithm for Multi-drone Path Planning and Coverage of Agricultural Fields. In Proceedings of the Agriculture Digitalization and Organic Production: Proceedings of the First International Conference, ADOP 2021, St. Petersburg, Russia, 7–9 June 2021; Springer: Berlin/Heidelberg, Germany, 2022; pp. 299–310.
46. Grlj, C.G.; Krznar, N.; Pranjić, M. A decade of UAV docking stations: A brief overview of mobile and fixed landing platforms. *Drones* **2022**, *6*, 17. [\[CrossRef\]](#)
47. Tian, R.; Cao, M.; Ma, F.; Ji, P. Agricultural UAV path planning based on improved A* and gravity search mixed algorithm. *J. Phys. Conf. Ser.* **2020**, *1631*, 012082. [\[CrossRef\]](#)
48. Zhou, H.; Xiong, H.L.; Liu, Y.; Tan, N.D.; Chen, L. Trajectory planning algorithm of UAV based on system positioning accuracy constraints. *Electronics* **2020**, *9*, 250. [\[CrossRef\]](#)
49. Mammarella, M.; Comba, L.; Biglia, A.; Dabbene, F.; Gay, P. Cooperation of unmanned systems for agricultural applications: A theoretical framework. *Biosyst. Eng.* **2022**, *223*, 61–80. [\[CrossRef\]](#)
50. Yingkun, Z. Flight path planning of agriculture UAV based on improved artificial potential field method. In Proceedings of the 2018 Chinese Control And Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 1526–1530.
51. Sun, Y.; Chen, W.; Lv, J. Uav Path Planning Based on Improved Artificial Potential Field Method. In Proceedings of the 2022 International Conference on Computer Network, Electronic and Automation (ICCNEA), Xi'an, China, 23–25 September 2022; pp. 95–100.
52. Kecman, V. *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*; MIT Press: Cambridge, MA, USA, 2001.
53. Staelin, C. *Parameter Selection for Support Vector Machines*; Technical Report HPL-2002-354R1; Hewlett-Packard Company: Palo Alto, CA, USA, 2003; Volume 1.
54. Elmaizi, A.; Sarhrouni, E.; Hammouch, A.; Nacir, C. A new band selection approach based on information theory and support vector machine for hyperspectral images reduction and classification. In Proceedings of the 2017 International Symposium on Networks, Computers and Communications (ISNCC), Marrakech, Morocco, 16–18 May 2017; pp. 1–6.
55. Mehta, A.; Ribeiro, E.; Gilner, J.; van Woesik, R. Coral reef texture classification using support vector machines. In Proceedings of the VISAPP (2), Barcelona, Spain, 8–11 March 2007; pp. 302–310.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.