Technology based on the Internet of Things to Monitor Animals

Jayaram Boga\textsuperscript{1*}, T.Sunitha\textsuperscript{2}, K.Manjula\textsuperscript{3}

\textsuperscript{1,2}Department of Computer Science and Engineering (Data Science), Marri Laxman Reddy Institute of Technology and Management, Dundigal, Hyderabad, Telangana, India.

\textsuperscript{3}Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology and Management, Dundigal, Hyderabad, Telangana, India.

Received: 14.06.2023 \hspace{1cm} Accepted: 22.06.2023 \hspace{1cm} Published Online: 30.06.2023

Abstract: Improving existing animal husbandry practices is essential before introducing grazing animals to vineyards. In order to provide this type of assistance, it is necessary to monitor and condition the animals' whereabouts and actions, especially their feeding posture. Using this strategy, sheep could graze in agricultural areas (such vineyards and orchards) without fear of harming them. Based on these findings, we have created an IoT-based platform for tracking animal habits. To facilitate unattended shepherding of ovine within vineyard areas, the system integrates a local Internet of Things network for data collection from the animals with a cloud platform with data dispensational storage competences. As a result, the system can tend to ovine flocks. Easy analysis and interpretation of Internet of Things (IoT) data is made possible by the machine learning capabilities built into the cloud platform. Therefore, we shall not only outline the platform but also supply some machine learning platform-specific results. To be more specific, testing looked at how well this platform could identify and characterize disorders related to animal posture. This page offers a comparison of the tested approaches because multiple algorithms were used.

Key words: IoT, Animal Monitoring, Big Data, Machine Learning, Posture Control.

*Correspondence: Assistant Professor, Department of Computer Science and Engineering (Data Science), Marri Laxman Reddy Institute of Technology and Management, Dundigal, Hyderabad, Telangana, India. Email: jayaramb@mlritm.ac.in

This work is licensed under a Creative Commons Attribution 4.0 International License CC BY-NC-ND 4.0.
1. INTRODUCTION
The cultivation of grapes, known as viticulture, is a rapidly expanding agricultural sector in Mediterranean countries. The situation in Portugal is illustrative because the country’s vineyards cover an estimated total area of around 191,000 hectares, making it the fourth-largest in Europe [1]. Increased demand for Mediterranean wines abroad is having a significant economic impact on the winemaking industry there. In order to prevent the wild species, sometimes known as weeds, from competing with the vines for sunlight and nutrients, they must be routinely removed in the spring. This is the most labor-intensive time of year for this task. Traditionally, weeding has been done by hand, though grazing sheep have been used in some locations. The vines and fruits, especially the lower branches, pose a threat to the human settlements because animals love to eat them. Since the pastures could not be used year-round, the sheep were the only animals allowed to graze there. Animal weeding has been phased out as part of the process of specialisation of winemaking activities that has occurred over the past fifty years. This, on the other hand, was automated in the blank spaces between the rows, and chemically applying herbicides to the spaces between the rows. Soil erosion can occur, and if it does, it will add to the emission of greenhouse gases caused by human-powered activities. The release of toxins might also poison food supplies and water supplies.

To preserve the sustainability of the livestock industry, the SheepIT project [2] seeks to identify strategies for luring sheep to eat the weeds that naturally occur in vineyards (and other comparable cultures) without harming the grapes or wine. This proposal uses a sensor network connected to the Internet of Things to keep track of where and how animals are standing within vineyards, with the goal of reducing the toll that livestock have on plant-based farms. The system also includes a cloud-based analysis platform, which is responsible for processing the collected data and deriving actionable insights. The processing power of today’s cloud services enables the combination of Data Mining and Machine Learning methods, allowing the extraction of fresh and useful information for whoever monitors vineyards and/or shepherds. With this data, vineyards and sheep ranches can be better administered. This article delves into the SheepIT cloud platform, which receives streamed data from animal sensors, performs data analysis (such as rule management), grants access to animal information, and sends instantaneous alarms upon detecting specific events (like panic attacks, predator attacks, an abnormal number of infractions, etc.). The SheepIT cloud platform takes in data streams from sensors on the animals, analyses the data (via methods like rule management), and provides access to relevant data about the animals. In addition, this study presents preliminary findings from an application in which several machine learning techniques are used to keep tabs on animal posture. What follows is a summary of the paper’s remaining content.

2. RELATED WORKS
Several papers on animal tracking have been written, each exploring a unique angle on the subject. The investigation of animal migration [3], the study of grazing animal behavior [4], the investigation of shaving site profiles [5], and the investigation of animal position and estrus [6], [7] are all very good
examples. Few studies really analyses data in real time; most focus instead on archiving sensor data from animal behavior monitors. Williams et al. [5] used machine learning techniques on GPS traces collected during the course of their investigation to categories the grazing habits of 40 cows across four months. They used the WEKA data mining programme and four different ML methods to conduct their study. Grazing, resting, and strolling were the three states they found. First, there was the grazing stage. The accuracy of location data is reduced when only one sensor is used to track an animal, particularly during the resting and grazing phases. The research proves that GPS receivers are to blame for the system’s excessive energy consumption, but it also shows that the technique may be applied to a number of different scenarios, such as activity tracking and disease prediction in animals.

Researchers focus heavily on estrus recognition [8] as a use case for animal monitoring. The potential economic gains from better insemination control are a major motivating factor. In order to monitor the behavior of the cows, researchers often attach accelerometer sensors to their necks or legs. When the information is analyzed, peaks in activity can be pinpointed as signs of estrus in the animal. Now, commercial options are available [9] that transmit monitoring data over a wireless network. This facilitates easier cow management by allowing for remote monitoring of feeding, ruminating, and activity patterns. Collars equipped with tri-axial accelerometry and magnetometry are used to collect data in the method proposed by Dutta et al. [10]. They are analysed by "Binary Tree" and "Naive-Bayes" classifiers, or companies of these classifiers, to determine cutoffs that are then used to categorise tasks. Classifiers as simple as the "Binary Tree" can sometimes achieve a 90% accuracy and sensitivity rate under ideal conditions.

Umstatter et al.[11] employed supervised behavioral categorization to classify the activities of a flock of ten sheep, identifying grazing, walking, standing, and lying down. The animals wore their GPS tracking collars while in a variety of settings, including on a hill, in level meadows, and within a shed. Maximum and minimum pitch also roll angles over 30 second aeras were utilized in the training data used for classification. Classification trees based solely on maximum pitch tilt achieved similar accuracies but were location-sensitive [6]. A manually developed rule set also achieved over 92% accuracy in classifying active and inactive behaviors. A well-crafted rule set, with an accuracy of over 95%, was used as a benchmark against which all of these techniques were evaluated. The final strategy only resulted in a 2% error rate when applied to the outdoor dataset.

3. A SYSTEM FOR MONITORING ANIMALS

The goal of the SheepIT project is to create a service based on the Internet of Things that will improve the monitoring and management of sheep flocks used for weeding vineyards. To do this, the system is constructed from a variety of modules, each with its own set of interconnected behaviours. Significant among these responsibilities is the requirement to collect data, compile that data, analyse that data, and communicate the results.

The key architectural components of the deployed platform are depicted in Figure 1. On the left, you
can see how the Internet of Things network architecture was put into action to complete neighborhood-level projects. The cloud platform on the right side of Fig. 1 can be connected to by the local infrastructure via a high-bandwidth connection, such as 3G, 4G, or LTE. This service is shown as being hosted online. We will delve more into these two foundational pillars in the sections that follow.

a. IoT network
The animals’ posture, activity, and whereabouts are monitored through the use of collars, which serve as the principal interface for gathering data from sensors. In addition, they relay information from the sensors to the computer. Where it may analyse sensor data and apply corrective stimuli (such as electrostatic and aural signals). This enables the programme to instantly analyses sensor data and implement adjusting stimuli. In the next step, the user data is sent to a permanent beacon network in the ground. In addition to gathering information from the collars, these units are also responsible for sending out synchronized beaconing signals at regular intervals across the network. The collars use these signals to pinpoint their exact location with RSSI-based methodologies, and the network uses them to monitor the animals’ whereabouts. The system employs a method called time division multiple access (TDMA), which is explained in depth in [6] along with the suggested solution for the system’s low power usage. There is an integrative node between a LAN and the Internet called "The Gateway." Because of this, the non-IP network can be easily combined with the IP-based Internet. Using the cloud platform discussed in this article could facilitate the deployment of such infrastructure across many, separately owned locations.

b. Cloud Platform
Figure 1 depicts the five parts of the Cloud platform that work together to collect, evaluate, and process stream data. These parts collaborate to acquire and examine stream information. An early landmark is the RabbitMQ Message-Oriented Middleware [7]. It’s supposed to make it easier for businesses to reach their customers. It receives the Gateway’s outgoing JSON messages and forwards them along to the data processing engine. The broker works with many different kinds of messaging protocols, including asynchronous publish/subscribe models like AMQP [12] and MQTT. Therefore, all messages generated by the Gateway are processed by RabbitMQ and stored in a FIFO queue before being made available to the rest of the platform. In addition to existing security measures, RabbitMQ may support SSL/TLS certificates.

At present, only Apache Spark has subscribed to a RabbitMQ queue, but other clients may do so in the near future. The platform’s primary processing structure is in charge of organizing the various services it provides. Data warehouse (DB) operations, JSON-to-DB conversions, alert generation, DM/ML-based data processing, and so on are all good examples. It combines a batch process for handling asynchronous traffic, such as that generated by platform-level processing activities, with a stream process for handling real-time data. Traffic created by platform-level processing activities is handled by the batch process, while real-time traffic is managed by the stream process. Both of these operations can create persistent data writes and schedule regular database updates. Adding Drools, a rule management module, to this processing architecture allowed for the creation of complex occasion
processing (CEP) also event stream processing (ESP). In a word, they consent us to make calculations, find designs, and zero in on other crucial links. May sound alarms, which may prompt the user to take immediate action.

![Figure 1. General layout](image)

The database is crucial to the operation of the whole system. Data collected by the collar’s sensors, information provided by the user that does not change, and other operational data should be stored in an easily accessible area so that higher-level applications can access it with minimal effort. Data collected by the collar’s sensors, or recovered with their help, is one example. Given that the SheepIT network has multiple nodes, a relational database structure seems to make the most sense. The anticipated message payload is not large enough to warrant implementing this architecture with a NoSQL model, despite the many benefits that come with doing so. PostgreSQL was selected from the available alternatives because of its scalability and the features it provides to keep data safe and secure. Finally, a REST API framework has been added to the platform to support web design. This allows you to not only communicate with the user, but also with any other platforms that could prove useful. Information about animals might be included in legally relevant databases like those used for animal registration.

4. ANIMAL BEHAVIOR MONITORING
Collars capture the majority of data on the provided platform. They’re in charge of keeping track of the sheep and recording data on their movements, postures, and actions. In addition to the hours of activity, the frequency of travel, the location and timing of preferred pastures, the occurrence of any anomalies (such as panic or illness), and the number of fence and posture infractions, the information gathered from the sensors can provide the farmer with a wealth of information. However, not all of this data can be collected by sensors; other methods, which require a lot of processing power, are required.
DM and ML methods, whose efficacy and popularity have been on the rise, give an intriguing potential that could be put to good use. Taking the platform’s machine learning features into account, we zeroed in on a particular, crucial use case. This software is used to analyse the stance of sheep.

A. Machine Learning Application: Identifying Posture Violations in Sheep

Since it is not always clear whether a sheep is snacking on weeds or vines, it takes a large number of sensors to rule out any false positives. Because of this, supervised ML algorithms were chosen to be used as a tool throughout the process. The goal of this type of machine learning is to build a model with data from a training set and then test it on new data to see how well it performs. The evaluation in this case is based on information gathered from collars equipped with a three-axis accelerometer and an ultrasonic transmitter. Neck pitch and ground clearance can be measured with the help of these collars. The sheep were filmed for three hours while they grazed freely on a flat pasture. The raw sensor data collected by collars at regular intervals was time-stamped and sent to a network for human classification.

![Figure 2](image)

**Figure 2.** resting(left), grazing(middle) and standing re caching for food examples

Therefore, we had a hard time categorising people into two categories. The more extensive initial classification will allow for subsequent classifications of activities that are more complex to reuse the same dataset, which may or may not be connected to the topic discussed in this article.

The ML strategies were implemented using the programming language R. The major purpose of the preliminary processing was to remove unnecessary details (as indicated by the entry’s sequence number and time stamp). This goal was reached. In the end, there were 20555 observations in the dataset, which were split evenly between a Training set of 15416 and a Test set of 5139. In the former case, algorithms are employed to define learning models, which are then evaluated, whereas in the later case, the models’ efficacy is verified. Models are put through their paces using both techniques.

A comparison of machine learning algorithms

The machine learning (ML) feature of the SheepIT computational platform was put through its paces in a number of studies designed to gauge the effectiveness of sheep posture violation detection. Since there are many ways to tackle classification problems, we looked at some of the more popular ones, including Random Forest, Decision Trees (DT) constructed using the C50 and rpart packages, XGBoost, K-
Nearest Neighbours (KNN), Support Vector Machine (SVM), and Naive Bayes. The findings were evaluated using several metrics typically utilised in classification jobs. The True Positive Rate (TPR) is a measure of how often a model incorrectly predicts a value when it should have been correct; it is also known as sensitivity. Measuring specificity with True Negative Rate (TNR) involves determining how often a model incorrectly predicts something it should get right.

Table 1. Algorithm Comparison For Machine Learning

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>ACC</th>
<th>TPR</th>
<th>TNR</th>
<th>PPV</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>0.9696</td>
<td>0.8267</td>
<td>0.9861</td>
<td>0.8728</td>
<td>0.987</td>
</tr>
<tr>
<td>DT(C50)</td>
<td>0.9693</td>
<td>0.8475</td>
<td>0.9833</td>
<td>0.8839</td>
<td>0.986</td>
</tr>
<tr>
<td>XGboost</td>
<td>0.9685</td>
<td>0.8267</td>
<td>0.9848</td>
<td>0.8625</td>
<td>0.988</td>
</tr>
<tr>
<td>KNN</td>
<td>0.9622</td>
<td>0.7702</td>
<td>0.9844</td>
<td>0.8503</td>
<td>0.977</td>
</tr>
<tr>
<td>SVM</td>
<td>0.9642</td>
<td>0.759</td>
<td>0.9879</td>
<td>0.8778</td>
<td>0.972</td>
</tr>
<tr>
<td>DT(rpart)</td>
<td>0.9591</td>
<td>0.8211</td>
<td>0.975</td>
<td>0.8728</td>
<td>0.97</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.9527</td>
<td>0.8795</td>
<td>0.9612</td>
<td>0.723</td>
<td>0.979</td>
</tr>
</tbody>
</table>

The Confusion Matrix is a result of evaluating a machine learning model on a dataset. This matrix is the foundation from which all of these metrics are derived. For the purpose of brevity, we have excluded several of the matrices from Table 1, although it does provide an overview of all the metrics that were considered for this analysis. The findings show that the algorithms are very similar to one another, however the highest levels of accuracy and AUC are generated by Random Forest, DT (with package C50), and XGBoost. More evidence of the precision of the results is shown in Figure 3, which shows that the curves overlap one another for the most part.

Creating the model and understanding its relationship between attributes and the classifier label (in this case, infraction or not) are two very separate tasks that need very different amounts of computer power. By allowing the model to be represented as a series of if statements followed by else clauses, DTs display the most relevant features in this context, allowing for fast model construction and tree-like interpretation. A label for an attribute is shown at the leaf, the last node on a branch. Each possible route from the root to a leaf defines a different classification rule, which can be expressed as a series of if statements. It is directly applicable to the SheepIT project since it is a highly useful method for acquiring the conditions necessary to construct a microcontroller-based sheep collar posture control mechanism. One of the most pressing problems that must be solved in vineyards is weed control. Winemaking is an expensive and time-consuming endeavour. Manufacturers who are looking for new ways to improve their products’ quality should avoid the tried-and-true methods of doing so, whether they be mechanical or chemical. Due to their propensity of munching on weeds, sheep are often seen as a more eco-friendly option. SheepIT, on the other hand, proposes a method that can condition the
posture and position of sheep while they graze in vineyards, protecting both the plants and the cultures that the sheep consume from injury. This system makes use of both local hardware and a cloud-based computing platform, with the latter responsible for gathering and processing information gathered on site. The initial data collection comes from the on-premises features. Machine learning is one potential mechanism of this kind that could be used.

5. CONCLUSION

From the collars worn by the sheep (which are mobile nodes) all the way up to the cloud platform, which is responsible for a broad variety of functions including data analysis, data processing, and data storage, the full architecture of the system is laid out in this paper. One of the most compelling examples of the platform’s potential is the identification of occasions in which sheep adopt postures that could endanger the vines and grapes. This scenario is a significant aspect of the SheepIT initiative. Using the preexisting framework, we were able to create and manage a data set made up of sensor data from the collar. We next proceeded to manually classify each data point in the set. The platform’s effectiveness was then assessed by evaluating its ability to run various machine learning algorithms. Particularly impressive is the interpretability of the results obtained through DT, which helped in the development of a posture control algorithm for collars. Despite the fact that all of the algorithms performed similarly in terms of accuracy, this was the case. Significant additional effort is required to assess the platform’s scalability and performance with larger datasets. There is also much left to accomplish to assess the platform’s usefulness for new machine learning use cases like sickness, panic attack, activity pattern, and food preference identification.
References


