EE305-Design Laboratory

Fog Computing To Detect Lameness

Automated Detection of Lameness in Animals using ML and Fog Computing approaches

INTRODUCTION

The Internet of Things (IoT) is rapidly becoming a thing. There are sensors everywhere, which also means that there is a lot of data being collected. Most of this data is analysed at centralised servers and in the cloud. The concept of Smart Dairy Farming is no longer just a futuristic concept, and has started to materialize as different fields such as machine learning have found practical applications in this domain.With the rise of IoT, cloud computing has become another popular move by major stakeholders. But then there's fog computing.

FOG COMPUTING

Fog computing is a term coined up at Cisco. Also known as fog networking/fogging, it is an architecture where some applications are managed at the edge of a network by a smart device. The rest of the applications and services are handled in the cloud. Fogging is essentially a middle ground between the cloud and the hardware to allow more efficient data processing, analysis and storage. This is achieved by reducing the amount of data to be transported to the cloud.

How Fogging Works

Fog Computing, or "fogging", is a distributed infrastructure in which certain application processes or services are managed at the edge of the network by a smart device, but others are still managed in the cloud. It is, essentially, a middle layer between the cloud and the hardware to enable more efficient data processing, analysis and storage, which is achieved by reducing the amount of data which needs to be transported to the cloud.

Difference between Fog and Cloud computing

IoT sensors and edge devices are usually where all the data comes from, the problem though is that these devices don't have the power to process these data. Cloud servers have this power and have been the ideal solution for this kind of situation. However, the servers are sometimes too far and data processing is not done in an efficient manner with regards to time. The process of transporting the data to these cloud servers also presents several security threats.

In a fog computing scenario, data is processed by a smart device therefore reducing the amount of data sent to the cloud servers. Fogging does not fully replace cloud computing, it serves to complement and make the data processing, analysis and storage more efficient. The advantage tha fog computing ads is that it brings intelligence to the local network.

Some of the Major Issues with Fog Computing

Authentication and Trust issues

Authentication is one of the most concerning issues of fog computing since these services are offered at a large scale. Fog service providers can be different parties like cloud service providers, internet service providers, and end-users.

Privacy

Privacy concern is always there when there are many networks involved. Since fog computing is based on wireless technology, there is a huge concern regarding network privacy. There are so many fog nodes that each end-user is accessible to them and because of this more sensitive information passes from end-users to the fog nodes.

Security

Fog computing security issues arise as there are many devices connected to fog nodes and at different gateways. Each device has a different IP address, and any hacker can fake your IP address to gain access to your personal information that is stored in that particular fog node.

Fog Servers

The right placement of fog servers should be there so that it can deliver its maximum service.The company should analyze the demand and work done by the fog node before placing it will help in reducing the maintenance cost.

Energy consumption

Energy consumption is very high in fog computing as the number of fog nodes present in the fog environment are high and require energy to work.Companies should try to minimize the energy requirement by the fog nodes so that they should become more energy-efficient and save costs.

Thus, Fog computing can handle massive data that arises from the IOT (Internet of Things) on the edge of the network. Because of its characteristics like low latency, mobility, heterogeneity it is considered to be the **best platform for IOT.**

APPLICATION OF FOG COMPUTING IN LAMENESS DETECTION

NEED OF LAMENESS DETECTION

The prevalence of lameness has been reported differently in different regions and states. Ger reported that on an average Irish farm, 20 in every 100 cows are affected by lameness in a given year. Timely detection of lameness is a big problem in the dairy industry which farmers are not yet able to adequately solve. It is one of the factors for reduced performance on many dairy farms, at least through reduced reproductive efficiency, milk production and increased culling . Lameness in sheep is the biggest cause of concern regarding poor health and welfare among sheep producing countries.

Best practice for lameness relies on rapid treatment, yet there are no objective measures of lameness detection.Lameness has many negative effects, including reduction in feed intake, reduction in milk production and weight loss. It therefore has a drastic effect on the performance of a dairy farm. Lameness is mostly detected at an advanced stage and thus requires immediate and often costly treatment. Health, welfare and production performance on farms can be improved by early detection and prompt treatment of diseased animals.

EXISTING APPROACHES TO DETECT LAMENESS

Although most farms are equipped with some kind of estrus detection system which is based on accelerometers, lameness detection systems based on the same have not been successful. This is because of vendor lock-in. Each of the systems would require its own hardware. Another assumption made by all the current solutions is that all the animals will get lame the same way irrespective of other factors such as season, location,etc. Also after some of the improvements, primary limitations of the previously proposed systems is that they follow the technique to process and analyse previously collected data and perform only cloud based analytics without leveraging and efficiently utilizing the resource which is not a useful system in the case of dairy animals where they need a regular check.The most prevalent techniques that were used previously to detect lameness include image processing, activity based techniques and through iot devices and data analytics.

PROPOSED MODEL

So now we have the challenge to detect lameness considering three important points-

(i) if we could detect lameness in sheep using both accelerometer and gyroscope signals,

(ii) which machine learning algorithms perform best at classifying lameness and what are the most important features for lameness classification

(iii) if we can classify lameness in sheep across a range of daily activities (walking, standing and lying).

Now a two phase approach is developed, aiming at classifying sheep lameness. The first phase classified sheep activity, distinguished between walking, standing and lying. Once a sheep's activity was identified, a second classifier was applied in order to classify if a sheep is lame or not.

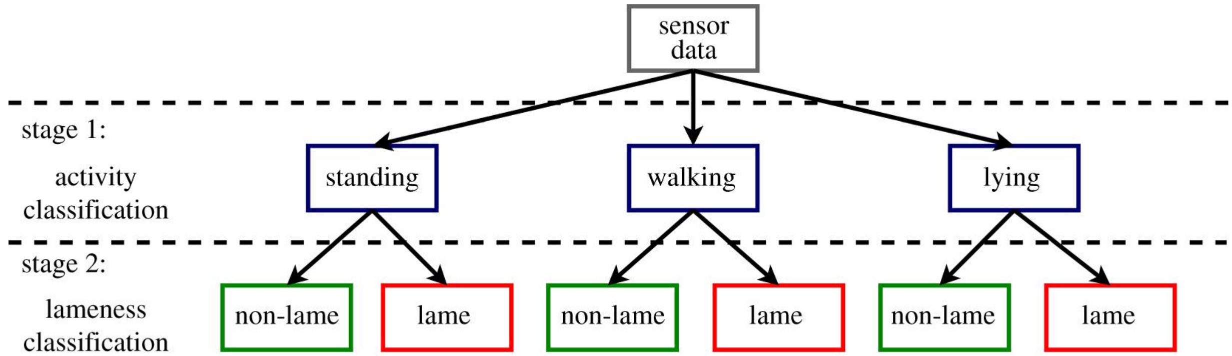


Fig. To illustrate the two phase classification approach

Now we'll compare algorithms that can differentiate lameness within three different activities (walking, standing and lying). 23 datasets (10 non-lame and 13 lame sheep) from an accelerometer and gyroscope-based ear sensor with a sampling frequency of 16 Hz is provided. The sample dataset contains the 32 feature characteristics for the sheep divided on the sheep Id and the day when it was collected over the gap of 7s. Sheep profiling to be done based on LMS score into two categories Lame and Non-Lame. After classification into walking standing lying we particularly used algorithms namely KNN(K nearest neighbours) and Random Forest (RF) to and compare the results and the accuracy obtained from both of them.

INSIGHTS INTO THE LAMENESS DETECTION ALGORITHM

So firstly, classified the sheep firstly on the basis of walking, standing and lying. Also to collect data statistics the given data is classified into lame and non-alme sheep ids. Then applied a feature selection process to order and rank features according to their importance. Feature selection in this study was carried out using a filter-based approach using ReliefF.

Once a sheep's activity was identified, a second classifier was applied in order to classify if a sheep is lame or not.

The classification algorithms which I have used to detect lameness are random forest (RF) and k-nearest neighbour (KNN). Both algorithms are implemented using the 'scikit-learn' package using the set of 32 previously described feature characteristics as input variables and the corresponding ground truth lameness and behaviour for each of the window samples as labels.

RF(Random Forest) Algorithm

In the RF algorithm, the number of trees was set to 250, with a minimum sample count set for leaf nodes equal to 1, and with the number of features to consider when looking for the best split equal to the square root of the *n* features.

KNN(K-Nearest Neighbour) Algorithm

For the KNN algorithm, *K* was set to 5, the distance metric was Euclidean and no normalization was adopted during the training as it did not significantly improve overall performance.

Then further an individual classifier model is developed within each of the different activities (walking, standing and lying). Classification performance was evaluated using 10-fold cross-validation, a commonly used methodology that provides a robust evaluation in classification model performance . This technique splits the original dataset into 10 subsets of equal size. Stratification was applied during the splitting to ensure that class representations in each of the subsets were equal to the original dataset. Stratification was based on the lameness class only. Then, over a total of 10 iterations (folds) are performed, and at each iteration, nine of the subsets were used to train a classification model, while the remaining one was held back as a test set. In each fold, the performance of the model built on the training set was evaluated using the test sets.

After the 10 folds, every one of the 10 subsets is used once as a test set, resulting in 10 sets of performance values, one for each fold. The average of these performance values represented the cross-validated classification performance. During the training of the data, no extra procedures were applied to balance the dataset as this was relatively well balanced for both activities and ratios for non-lame and lame samples in the three different activities.

DEPLOYMENT DETAILS

Google Collab

https://colab.research.google.com/drive/1UxDmLAupk2b1VM36bxP7v2Qb-fPCfLFf?usp=sharing

Source Code

https://github.com/chanmol1999/LamenessDetection

PERFORMANCE OF THE CLASSIFICATION

The performance of the best performing algorithms out of two is evaluated using the metrics overall accuracy, precision, recall (also known as sensitivity), *F*-score and specificity as follows:

$$\text{overall accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}},$$

$$\text{precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}},$$

$$\text{recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},$$

$$F - \text{score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

$$\text{and} \qquad \text{specificity} = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}},$$

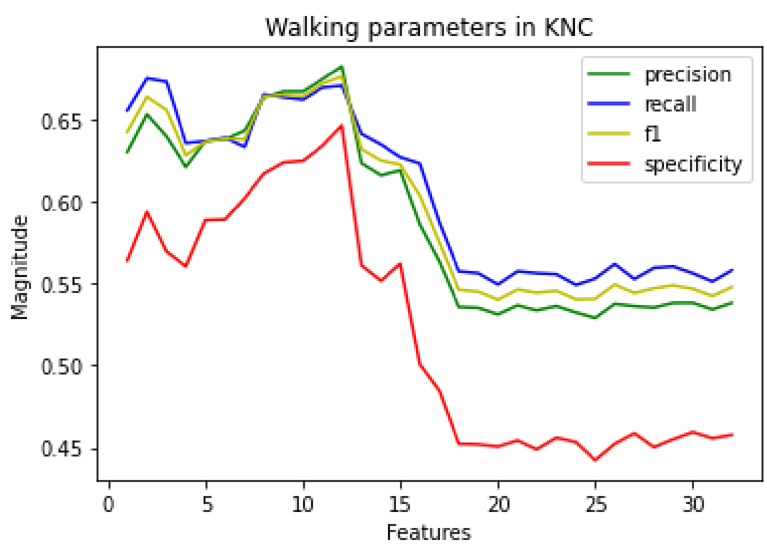
where TP (true positives) represents the number of instances where lameness was correctly classified by the algorithm and visually observed. FN (false negatives) represents the number of instances where lameness was visually observed, but was incorrectly classified as non-lame by the algorithm. FP (false positives) is the number of instances predicted as lame by the algorithm but observed as non-lame. TN (true negative) is the number of instances where the algorithm correctly classified a sample as non-lame when it was actually observed as non-lame. The F-score gives a measure of a test's accuracy and is the harmonic mean of precision and recall.

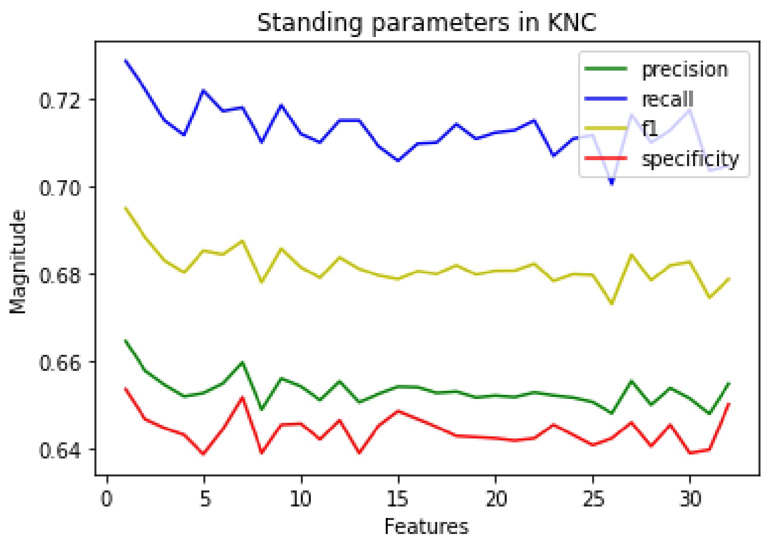
RESULTS

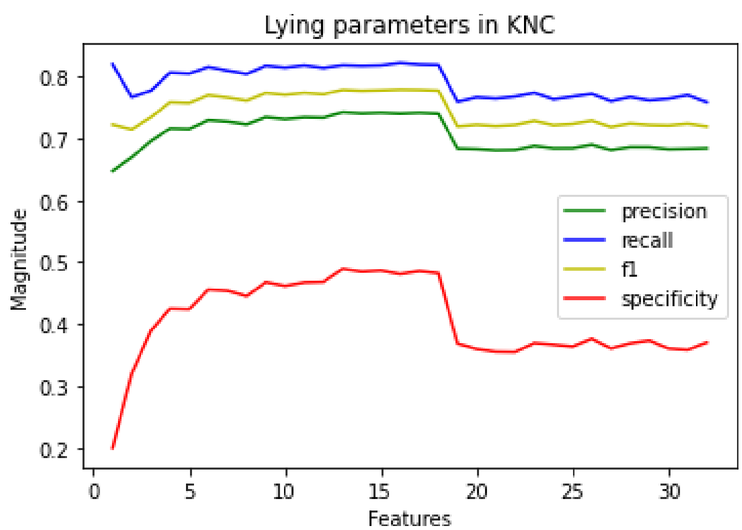
Facts Collected from Dataset

The ids of the sheep that are identified as non lame are - 1,2,3,4,10,11,12,15,16.The ids of the sheep that are identified as lame are d- 6,7,8,9,13,18. Also the data is given in the discrete form of the days such as 1,6,7,14,17,31 accordingly for lame and non-lame each in mean of 7s duration. In total, there were 20104 samples, each 7 s in duration. Of these samples, 31.57% (46.88% non-lame and 53.12% lame) of the samples included walking, 36.87% (51.56% non-lame and 48.44% lame) included standing and 31.56% (35.78% non-lame and 64.72% lame) included lying activity. Classification of the dataset into lame and non-lame profiles can be done on the basis of LMS score where 0,1 for non-lame and 2,3 for lame animals. Currently only the accelerometer and gyrometer have been used for classification which results in 32 feature characteristics i.e., 16 from each.

KNN Algorithm Performance across Activities

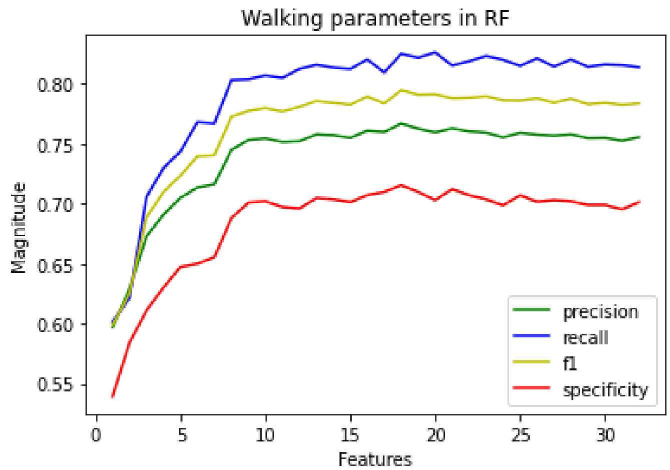


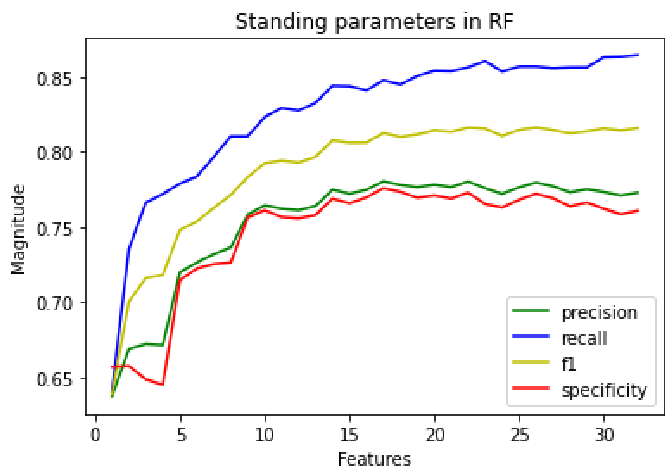


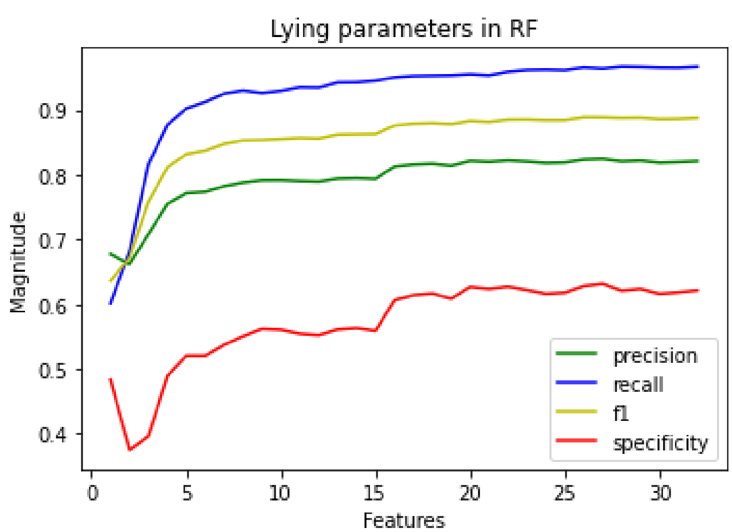


Graph Insights- For walking, all the different performance metrics of the classification increased with an increased number of features between 1 and 11 after which they decreased After an initial increase, performance metrics did not show some significant changes. For standing, similarly all the different performance metrics of the classification remained almost constant with an increase in number of features. For lying, precision recall and F-score metrics of the classification increased with an increased number of features between 1 and 15 at which point all of them plateaued with a small decrease. Of all the performance metrics, specificity was the lowest and recall was the highest across all activities.

RF Algorithm Performance across Activities

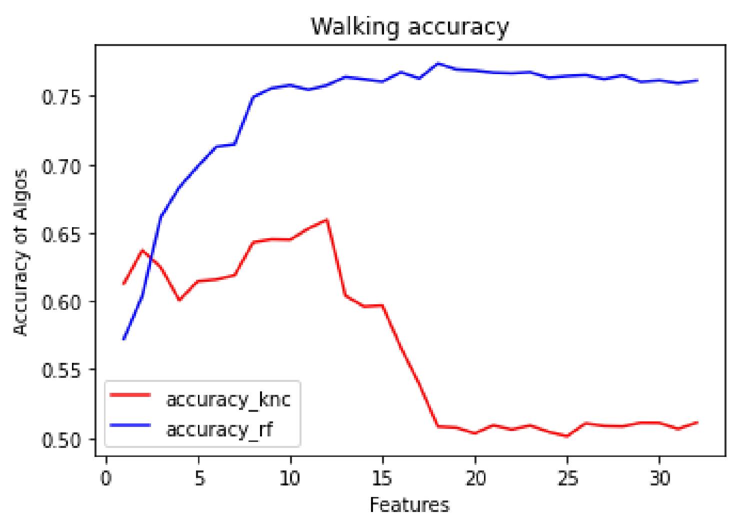


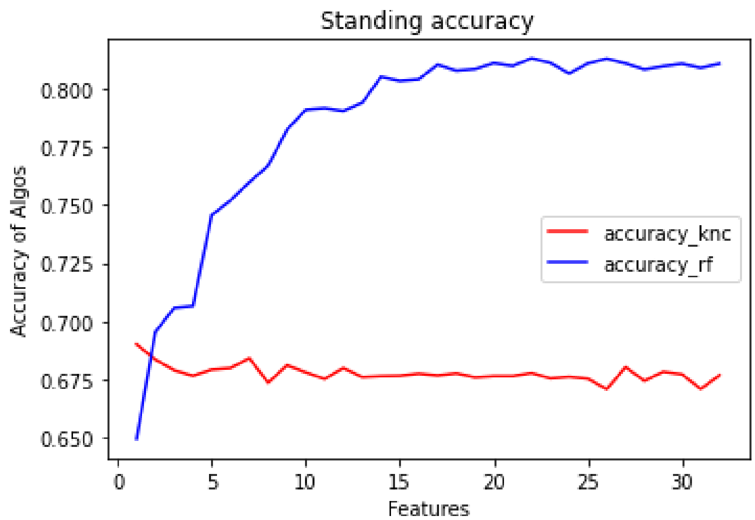


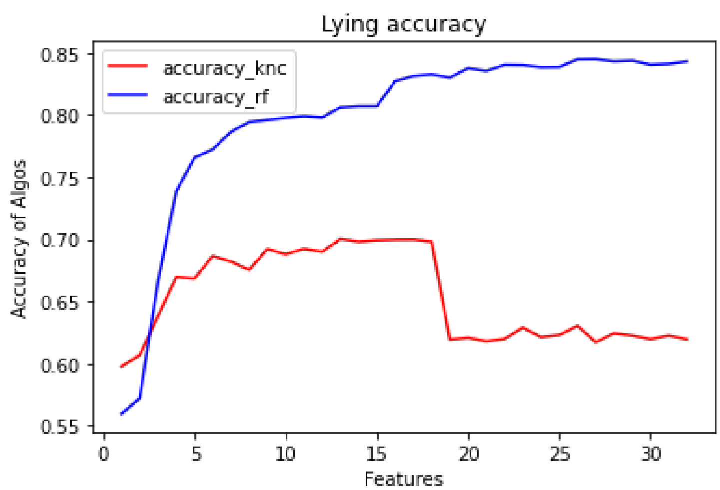


Graph Insights - For walking, all the different performance metrics of the classification increased with an increased number of features between 1 and 8 at which point all plateaued. After an initial increase, performance metrics did not show any significant changes. For standing, similarly all the different performance metrics of the classification increased with an increased number of features between 1 and 9 at which point all plateaued. For lying, precision recall and F-score metrics of the classification increased with an increased number of features between 1 and 5 at which point all of them plateaued with a small increase. Maximum performance values were obtained when using features 22, 32, 22 and 22 for precision, recall, F-score and specificity, respectively. Of all the performance metrics, specificity was the lowest and recall was the highest across all activities.

Comparison of Accuracy of Algorithms across Activities







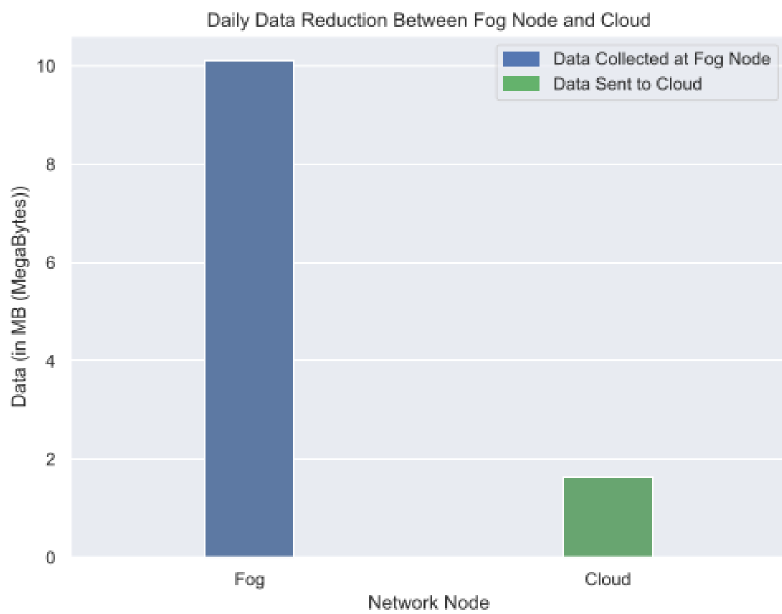
When classifying lameness in walking, RF obtained the highest overall accuracies, followed by KNN, when considering all activities. Performance of KNN deteriorated as more features were added to the classification. The best accuracy, in an unconstrained scenario, was obtained using RF and 17 features, yielding an overall accuracy of 76.83%. In terms of overall accuracy, the RF algorithm started to plateau when more than eight feature characteristics were used, yielding overall accuracies between 74 and 76%.

Overall accuracy in standing is generally higher than the ones in walking. Once again, RF consistently outperformed the KNN with accuracy peaking at 81.15% when using 22 features. Accuracy started to plateau between 80 and 81% once 15 or more features were used.

Classification of lameness within lying provided the highest accuracies compared to the other two behaviours, with RF yielding the best accuracy at 27 features with a value of 84.91%. Apart from the global maximum at 27 features, RF accuracy plateaued after 17 features between 83 and 84%. The difference between RF and KNN performances overall was larger than those seen for walking and standing.

FOG COMPUTING APPROACHES

We observed that on average our model had an accuracy of 80-85%. So now only that data needs to be sent to the cloud that contradicts our prediction and needs further processing, i.e. through this approach we reduced data transfer to the cloud.



With this approach we can build custom models for small groups of animals in the herd that share similar features within the herd. This improves the accuracy of the lameness detection as opposed to a one-size fits all approach which is practically not possible due to various factors such as location. Fog-based computational assistance enables the intelligent processing of data closer to the source, thereby leading to an 80-85% reduction in the amount of data transfer to cloud. By utilizing fog architecture we will be able to reduce data exchange between fog and cloud nodes from 10.1 MB to 1.61 MB on a daily basis. As shown in the figure, an 84% reduction in the amount of data that would otherwise have streamed to the cloud throughout the day. This aspect of data reduction becomes even more crucial while scaling up the farm and the herd, as the amount of data collected and streamed would then rapidly increase. Another key lesson learned is that any of the edge/fog/cloud resources of the overall architecture if considered in isolation would not be able to manage the developed IoT application, without compromising on functionalities or performance. And thus a careful coordination of edge, fog and cloud components is required as presented in this work.

FUTURE WORK

An ML model has been developed with the pre-processing of the data which ultimately depends on the local animal characteristics collected through IOT devices. Goal of the project involves deploying the model at a fog node after training offloading it into the cloud and only relevant information will be stored in the fog. Fog will serve as a pre processing and analysis unit.One of the possible directions of future work is to look into distributed learning and distributed data analytics based approaches in such real-world IoT based deployments. Once the developed technology has been validated on a number of farms with different geographical and environmental settings, the goal is to roll out the technology to the vendors' customer base as an added feature through licensing.May be An app to notify farmers regarding lame animals making it less dependent on internet connectivity.

REFERENCES

- Mohit Taneja, John Byabazaire, Nikita Jalodia, Alan Davy, Cristian Olariu,Paul Malone, 2020. *Machine learning based fog computing assisted data-driven approach for early lameness detection in dairy cattle*. www.elsevier.com/locate/compag

- Jasmeet Kaler, Jurgen Mitsch, Jorge A. Vázquez-Diosdado, Nicola Bollard, Tania Dottorini and Keith A. Ellis,15 Jan 2020. *Automated detection of lameness in sheep using machine learning approaches: novel insights into behavioural differences among lame and non-lame sheep.*https://doi.org/10.1098/rsos.190824

- Miiaková, M., Strapák, P., Szencziová, I., Strapáková, E., Hanušovský, O., 2018. *Several methods of Estrus detection in cattle dams: a review. Acta Universitatis Agriculturae Et Silviculturae Mendelianae Brunensis 66(2),619–625.* https://www.researchgate.net/publication/324902783_Several_Methods_of_Estrus_Detection_in_Cattle_Dams_A_Review https://acta.mendelu.cz/66/2/0619/https://doi.org/10.11118/actaun201866020619

- Van Nuffel, A., Zwertvaegher, I., Van Weyenberg, S., Pastell, M., Thorup, V.M., Bahr, C., Sonck, B., Saeys, W., 2015. *Lameness detection in dairy cows: Part 2. Use of sensors to automatically register changes in locomotion or behavior. Animals.* https://doi.org/10.3390/ani5030388.

- https://towardsdatascience.com/an-implementation-and-explanation-of-the-random-forest-in-python-77bf308a9b76

PREPARED BY

ANMOL CHADDHA

1801EE07

B.TECH. ELECTRICAL ENGINEERING, IIT PATNA

(BATCH OF 2022)

chanmol1999@gmail.com