Real-Time Mastitis Detection in Livestock using Deep Learning and Machine Learning Leveraging Edge Devices

by

Kawshik Kumar Ghosh
19101057
MD.Fahim-Ul-Islam
19101294
Abrar Ahsan Efaz
19101368
Md. Wahid Sadiq Ratul
19101194
Md Zaid Hassan Khan Shatil
18201192

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
School of Data and Sciences
Brac University
January 2023

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Kawshik Kumar Ghosh

19101057

MD.Fahim-Ul-Islam

19101294

Abrar Ahsan Efaz

19101368

Md. Wahid Sadiq Ratul

19101194

Md Zaid Hassan Khan Shatil

18201192

Approval

The thesis/project titled “Real-Time Mastitis Detection in Livestock using Deep Learning and Machine Learning Leveraging Edge Devices” submitted by

1. Kawshik Kumar Ghosh (19101057)
2. MD.Fahim-Ul-Islam (19101294)
3. Abrar Ahsan Efaz (19101368)
4. Md. Wahid Sadiq Ratul (19101194)
5. Md Zaid Hassan Khan Shatil (18201192)

Of Fall, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 19, 2023.

Examining Committee:

Supervisor:
(Member)

____

Amitabha Chakrabarty, PhD

Associate Professor
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

____

Md. Golam Rabiul Alam, PhD

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

____

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

The livestock industry is a vital component of the global economy, with a value estimated at over $1.4 trillion. However, the health of livestock animals is frequently threatened by infectious diseases, which can have serious consequences for the industry and the economy. Bovine mastitis is one such disease that is prevalent and costly to treat. It is caused by bacterial infection of the mammary gland in cows and can have severe impacts on the dairy industry. In developing countries like Bangladesh, where the livestock sector is a significant contributor to the national economy, mastitis is a major concern. It is estimated that this disease costs the dairy industry millions of dollars each year in Bangladesh, due to reduced milk production, increased treatment costs, and culling of infected animals. The economic impact of mastitis can be particularly significant in a country like Bangladesh, where the livestock sector plays a vital role in the economy. In order to overcome this issue, this paper presents a real-time system for detecting mastitis in livestock using Deep Learning and Machine-Learning techniques leveraging edge devices. The proposed system aims to provide a timely and accurate diagnosis of clinical mastitis, ultimately reducing costs and improving the efficiency of treatment. By utilizing deep learning and machine learning techniques, the system is able to analyze data from edge devices and make accurate predictions about the presence of mastitis. This can help farmers and veterinarians identify infected animals and take appropriate action to prevent the spread of the disease. In the proposed system, various Deep Learning and Machine Learning algorithms were utilized for classification, and a comparison was made based on their accuracy and performance. The models that performed best with the highest accuracy were selected for further use. InceptionV3 and Random Forest algorithm were chosen for Deep Learning and Machine Learning, respectively, and had an accuracy of 99.34% and 99% respectively. A review of other papers that have used classification techniques for detecting mastitis shows that the models proposed in this paper have demonstrated better accuracy in the diagnosis of mastitis in livestock. The real-time system for detecting mastitis in livestock presented in this paper has the potential to significantly reduce the economic impact of this disease in the dairy industry of Bangladesh and other developing countries. By providing a timely and accurate diagnosis, the system can help to improve treatment efficiency and protect the health and productivity of livestock animals. In doing so, this system can have positive impacts on the livestock industry and the global economy by improving the health and productivity of livestock animals and reducing the costs associated with mastitis.

Keywords: Deep Learning; Machine Learning; Edge Devices; Mastitis; Livestock

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our supervisor Dr. Amitabha Chakrabarty for his kind support and advice in our work. He helped us whenever we needed help.
And finally to our parents without their support it may not have been possible. With their kind support and prayer, we are now on the verge of our graduation.

Table of Contents

Chapter 1

Introduction

Livestock production is essential to the global economy, with over 30% of the Earth's ice-free land area dedicated to this purpose and an estimated value of $1.4 trillion [12]. Additionally, the livestock industry is a major source of employment, with 1.3 billion people globally depending on it for their livelihoods, and providing direct support to 600 million poor rural household farmers in developing countries [12]. In Bangladesh, it is also a notable contributor to the country's GDP, with a 6.5% contribution [24]. However, the success of any cattle enterprise is affected by the presence of infectious illnesses caused by bacteria, viruses, rickettsia, and fungus. One of the most common and expensive diseases affecting livestock is bovine mastitis, a bacterial infection that affects the mammary gland of the infected animal [24],[32]. Due to the deterioration of milk quality and the infectious nature of the disease, bovine mastitis significantly harms the economy of the food industry [22]. There are two categories of mastitis which are Clinical Mastitis and Sub Clinical Mastitis. There are no external symptoms for Sub Clinical Mastitis whereas there symptoms for Clinical Mastitis [19]. On average, the disease costs an estimated $147 per cow annually in failure costs [32]. In Bangladesh, the disease is responsible for the huge loss in the dairy industry with an annual economic loss of Tk 122.6 million or US $2.11 million [24]. Therefore, the disease has a serious adverse effect on the economy of a developing nation like Bangladesh. There are various conventional techniques for identifying mastitis at an early stage, including bacteriological culture and somatic cell counting [22]. Still, these procedures take time, call for technical expertise, and necessitate a laboratory [22], [21].

The paper presents a real-time system for detecting clinical mastitis in livestock using deep learning and machine learning techniques. The system aims to provide an accurate and timely diagnosis of mastitis, which ultimately reduces costs and improves the efficiency of treatment. The proposed system utilizes edge devices to gather data, which is then analyzed by deep learning and machine learning techniques to make accurate predictions about the presence of mastitis. This can aid farmers and veterinarians in identifying infected animals and taking appropriate action to prevent the spread of the disease. In this paper, the dataset contains numerical data on the health of 1100 cows and the corresponding milk images which were taken over a period of six days. The image is divided into two groups—normal and abnormal to aid in the diagnosis of this illness. The input data was initially augmented in order to increase the number of abnormal pictures of infected cows with mastitis sickness. Regarding the data multiplication, input data is divided into three categories (training, testing, and validation) to verify accuracy and acceptable picture characteristics. Three well-known convolutional neural network-based models are used which are Resnet50, InceptionV3, and VGG16. The model is trained for performance monitoring, and the findings are verified using these classification techniques. Out of the three models, inceptionV3 was selected for further use as it provided the best accuracy at 99.34%. The numerical data is classified into two classes, 0 and 1 which represent healthy and mastitis-infected cows respectively. Six different well-known classification algorithms were used for the classification. The algorithms that were used for the classification are Random Forest, K-nearest Neighbors, Support Vector Machine, Decision Tree, Logistic Regression, and Naive Bayes. Out of the models, Random Forest was selected for further use as it had the highest accuracy at 99%. The models are then deployed and compared through the use of an edge device. Through the comparison of both models, the final result is computed and displayed. The real-time system for detecting mastitis in livestock presented in this paper has the potential to significantly reduce the economic impact of this disease in the dairy industry of Bangladesh and other developing countries. By providing a timely and accurate diagnosis, the system can help to improve treatment efficiency and protect the health and productivity of livestock animals. This system can have positive impacts on the livestock industry and the global economy by improving the health and productivity of livestock animals and reducing the costs associated with mastitis.

1.1 Research Problem

The livestock sector is a vital global asset valued at $1.4 trillion and supports the livelihoods of 600 million families worldwide [12]. Livestock products constitute 17% of world calorie consumption and 33% of international protein consumption. In developing countries like Bangladesh, the livestock sector is a significant contributor to the economy, accounting for 6.5% of GDP and providing livelihoods to 85% of the population who are directly or indirectly engaged in agriculture [24] [26]. The country has 24 million in cattle and 1.4 million in dairy farms, with cows being the primary source of milk production [26]. The health of livestock is crucial for the success of the livestock industry, however, infectious diseases caused by various microorganisms pose a threat to the health of the livestock, which in turn has a direct impact on the economy. Bovine Mastitis is one of the most widespread and expensive diseases affecting the livestock sector, with an estimated cost of Tk 122.6 million or US $2.11 million annually in Bangladesh alone [24]. The disease is caused by inflammation of the mammary gland through bacterial infection [23]. With a 13.3% prevalence of clinical Mastitis, the disease is a major concern for the dairy industry in Bangladesh and is the leading source for the use of antibiotics in dairy farms [24]. While conventional methods such as somatic cell counting and California Mastitis Test (CMT) are available for detecting Mastitis, they are time-consuming and require technical expertise or laboratory facilities [21]. This paper proposes an alternative and more efficient method of detecting Mastitis in real-time at an early stage through the use of Deep Learning and Machine Learning algorithms utilizing edge devices.

1.2 Research Objectives

Through our proposed system, we aim to detect Mastitis in cattle in a more efficient and cheaper way than most traditional methods which will not require the use of any technical skills or laboratory and will, in the long run, be beneficial for the growth of the economy. The objectives of this research are:

- To compare several deep learning models to evaluate the model with the best accuracy.
- To present a real-time system for detecting bovine mastitis in livestock using Deep Learning and Machine Learning techniques.
- To provide an accurate and timely diagnosis of bovine mastitis to reduce costs and improve the efficiency of treatment.
- To utilize edge devices to gather data and analyze it through deep learning and machine learning techniques to make accurate predictions about the presence of bovine mastitis.
- To aid farmers and veterinarians in identifying infected animals and taking appropriate action to prevent any further spread of the disease.
- To evaluate the potential of the real-time system to significantly reduce the economic impact of bovine mastitis in the dairy industry of Bangladesh and other developing countries

Chapter 2

Literature Review

The use of Machine Learning and Deep Learning models for the detection of diseases among livestock has had a big impact on the predictability and avoidance of any potential herd-level livestock infection. Diseases like Bovine Respiratory Disease (BRD) which is responsible for 640 million dollars annually just in the United States can be predicted to an extent by the use of IoT sensors and classification algorithms. In [9] for bovine tuberculosis (bTB) GBT, a machine learning classification algorithm had better accuracy at predicting btB breakouts than traditional SICCT tests.

Moreover, Machine Learning or Deep Learning models have been deployed on various edge devices that assist in the real-time detection of several diseases and have been used for real-time health monitoring of livestock animals which has had huge success at predicting any disease that cattle might face. In [1], experiments on disease detection systems were done with an accuracy of 90%. Different physical parameters including heart rate, temperature, humidity, voice, and image are collected using different sensors and used as data in the classification system to give an efficient accuracy at detecting and predicting potential harmful diseases at an early stage and thereby assisting in the overall livestock industry.

2.1 Machine Learning/Deep Learning Classifiers

Deep Learning and Machine Learning techniques have drastically enhanced performance in a wide range of advanced visualization and classification applications, involving disease detection, object detection, and monitoring. A classification model with an accuracy of 90% was created using Decision Tree, in paper [1]. Here, the decision tree uses the training data given in tabular format consisting of different classes and attributes to make a tree that has nodes and to know each position of these nodes, Entropy E (s) value was calculated. CNN (Convolutional Neural Network) was used to identify animal illnesses in papers [2], [4], [5], and [14]. In paper [2], The researchers developed a new Convolutional neural dairy motion sensor as a neckband on the livestock to gather reliable data of cough and sneeze along with temperature, moisture content, behavior, and atmospheric value systems in order to convert the analog audio characteristics to digital where MFCC is being used for noise harvesting as it needs to perform well with k-NN. Nearest neighbor is performed to the sound in order to classify everything within the database's closest pattern. The obtained result by the proposed approach was being used to compare two dynamical patterns. Consequently, if the current cough nerve impulses meet or lie within the similarity extracted features of the classification algorithm, the neural network provides additional variables (ecological, biological, and many others) to identify the characteristics of the speech signals for BRD characterization. In the study [9], the authors tried to get a better accuracy at predicting bTB by using ML algorithms than traditional SICCT test where the authors experimented with four different classifier algorithms and continued with Gradient Boosted Trees (GTB) as it showed better results. The authors as a database worked with the bTB surveillance data (SAM) containing the positive and negative SICCT results, including SICCT test results of herds, land cover and climate data. Later GTB showed better accuracy results than SICCT tests. In the study [12], a deep learning model was proposed using an IoT based Livestock monitoring system which detects several physiological parameters of livestock animals and predicts cattle health overtime using Artificial Neural Network (ANN), Support Vector Machine (SVM), and Random Forest (RF), Linear Regression Model. Support Vector Machine (SVM) was used in article [4] for object categorization of animal diseases and a dataset with 1200 images was used in the training set. Throughout research article [15], Machine learning techniques has been extensively proposed in the literature over last decade. Sensors, such as webcams, are used to attain a high level of accuracy. The researchers define cow behavior patterns using supervised machine learning practices. They also utilize theoretical concepts to detect the psychological states of calves. A wristwatch application was developed to collect remotely sensed data from the forearms of horses and riders. The acquired set of data was loaded into ML classification models (Decision Tree, NN, Naive Bayes, Linear Discriminant Analysis and k-Nearest Neighbors). Cyclic voltammogram (CV) was used in the study [3], where an efficient and less time-consuming approach to detect BRD was approached by the author and for that an Ultramicroband electrochemical sensor was used. The ultramicroband was first initially changed using a captured bimoleculespecific to the virus or antibody. Using Cyclic Voltammogram (CV) of 1um microband electrochemical sensor with 1 Mm FcCOOH for pre-test, negative BVD virus serum was detected and results that were achieved were accurate. In the study [10], deep learning methods are used in both local and cloud servers and 2D signal processing algorithms were used to allow for state-of-the-art thermographic analytical methods. Linear regression and polynomial regression are used to monitor cattle body temperature and diagnose common illnesses such as foot-and-mouth disease detection and rumen acidosis disease detection with an accuracy of 63.87%, [6]. In article [5], the system employed in this study is built on a service-based approach that is free of vendor lock-in and scalable, allowing for the easy addition of additional features such as calving, heat, and lameness difficulties. This study used one Japanese Black Cow and one Holstein cattle. Environmental temperature, humidity, rumen temperature, and deep body temperature were measured using an infrared camera to capture images, an ML-based system to monitor temperature data, and RFID (Radio Frequency Identification) tags with Ultra High Frequency (UHF) band combined with built-in temperature sensors. Scikit-learn, Linear regression, and Quadratic Polynomial regression models were utilized to process the data. Environmental temperature and humidity are input parameters. This results in the production of deep body temperature. In the [11], the publishers primarily proposed a methodology for examining the communication method for an in-body to the out-of-body arrangement in wildlife in research [11]. They conduct trials on a cannulated cow with its sensor put inside the animal's rumen at 40,000,000Hz. Motion tracking cameras are also utilized to build a 3d representation of the cow, which is then used to assign probabilities. These modeling results demonstrate route loss of equivalent size to the tests. The proposed scheme can analyze the routing protocol for different animals at any frequency. Aside from paper [8], the overall structure requires hardware: A static reader linked to an interaction terminal; a fixed transmitter and a transmitter positioned at the stables' entry for learning to read and write chip; a transportable PC connected to a remote database; and a programmable logic controller attached to an electronic tag. Research has evaluated the efficiency of an animal authentication and identification system based on the usage of "read and write" embedded sensors packed in envelopes that may be placed in the animal. The labels' memory is split into two categories: one that only displays the animal's ISO-code and cannot be modified (read-only type), and the other that may be changed to update information it included. The KDE method, a Moving mean-based algorithm, was used in [7] to demonstrate the possibility of a revolutionary automatic system for identifying and tracking cows in broad livestock systems using space-time data given by a low power global positioning system (LP-GPS). In [16], novel sensing and interference algorithms, and data aggregation methods were employed to diagnose Mastitis in farm animals with great accuracy under correct conditions. In paper [17], this research describes an automated, IoT-based dairy cow monitoring system. The system is comprised of hardware devices, a cloud infrastructure, an end-user application, and cutting-edge data measurement and analysis algorithms. The system was tested in a real-world scenario and passed. To capture animal activity in this study, a specialized battery-powered microcontroller combined with low-powered inertial sensors and wireless communication was employed. This gadget collects data on rumination, eating, walking, and other activities in order to determine estrus. HUB serves as a gateway for data to be received and stored on a cloud server. The Raspberry Pi Zero W single-board computer was used to develop the HUB. An android application is also used to display data for each individual animal. A comparison of several classification algorithms was done in paper [18] to get the best results for detecting mastitis. J48, Random Forest, Support Vector Machines, k-nearest Neighbor, and Naive Bayes algorithm were used on a Mastitis dataset containing 400 instances and 7 attributes taken from 100 cows on a monthly basis. All the algorithms had an accuracy of over 90% and the best performance was an accuracy of 98% achieved using the J48 algorithm. In [19] a similar study was done with the same datataset, that they had collected through smart sac device. They used KNN and SVM for the classification and had an accuracy of 73.33% and 86.66% respectively. A comparision using machine learning algorithms to predict udder health was done in [20] which had an accuracy above 75%. A convolutional neural network called InceptionV3 is employed in [31] which has top-class accuracy and is incredibly adaptable and precise. The number of parameters in the model can be drastically reduced by effectively breaking the network down into smaller convolution sections [25]. The network model makes use of three separate Inception modules, each of which combines convolution sections of various sizes [27]. The reason it was selected to be utilized as a model is due to its flexibility and capacity to deconstruct features into smaller convolution sections [31]. In paper [25], InceptionV3 and InceptionV4 were used to identify obscured ships where InceptionV3 had a better training time.

Another convolutional neural network named VGG16 has been used in the paper [28] where the research provides a unique technique for using deep convolutional neural networks to integrate aligned RGB and NIR pictures captured by RGB-D sensors for fruit recognition. In the paper [29], the proposed model with VGG16 gave results that showed a precision of 92% in detecting COVID-19 with chest CT images. In the paper [30], for the detection of COVID-19, the VGG16 transfer learning model was used where the proposed model provided a recognition accuracy of 99.5%.

2.2 Devices and Sensors

Several devices and sensors were employed to detect a variety of physiological characteristics in order to collect real-time data from livestock animals, which was then used as training data for machine learning and deep learning categorization. A square shaped strap attached box with an ESP 32, Arduino Mini Pro including temperature and heart rate sensors was used in paper [1] for collecting the real time data from the livestock animals for data classification. Similarly, in [2] and [12] temperature, heart rate, humidity sensors and camera for live monitoring of the cattle was implemented. In [2] microphone was used in the device to record real time audio of the cattle. Accelerometer was used to track and record cattle movement and behavioral changes and collect data in IOT devices featured in papers [2], [4], [12], [13] and [17]. GPS tracking devices were also used in paper [12] and [16] for the same purposes. Ultramicroband electrochemical immunosensor was used in [3] for Early detection of Bovine Respiratory Disease through a system that detects the presence of BRD virus in blood samples of cattle in a farm using Cyclic Voltammogram (CV). Cloud servers were used to store transferred data to databases for analysis and processing in devices featured in papers [5] and [10]. Use of RFID for processing the data of farmers and movement of the cattle was seen in paper [8]. In [6], IR cameras were used to detect the temperature including other temperature/humidity/illuminance sensors. On [11], a Frequency synthesis device (EV-ADF4355SD1Z) with an ANT500 transmitter and an alkaline battery of 9V was being used to test this hypothesis in a dairy farm with a cannulas cow for which the connector was wrapped (inset) and injected into the cow's rumen via the cannula. Low power global positioning system (LP-GPS), LPGPS system, SigFox, power supply, AppWeb, GIS software tool, Kernel Density Estimation models was used in [7] to demonstrate the viability of a unique automatic system based on space-time data for detecting and tracking cows in large livestock operations received from the devices. NB-IoT network was used in neck-mounted sensors for both captive breeding and grazing breeding in [14].

The below table 2.1 contains the summary of the literature review:

Ref.	Task	Devices	Algorithm	Dataset	Accuracy
[1]	Disease detection and Health monitoring system	Strap attached box with sensors.	Decision Tree Algorithm	Manually collected sensor data	90%

[2]	Bovine Respiratory Disease diagnosis	AI infused cow necklace.	Dynamic Time Warping (DTW), nearest neighbor (NN) algorithm.	Cattle audio from dairy farms	Cosine angle 0.89
[3]	Detection of Bovine Respiratory Disease within blood samples.	Ultramicro-band	Cyclic Voltammogram (CV)	Blood samples	1.2 nA to 1.4 nA.
[4]	Animal Disease Diagnosis Expert System	Accelerometer sensors	Image classification algorithms, 4 Machine Learning algorithms.	Dataset of 1200 images	97.06%
[5]	Disease detection in dairy farming.	Receiver and Transceiver, IBM Watson IoT platform	Fog Node Using Constraint Programming	Fog Node data	Graphical representation
[6]	Livestock disease detection.	IR cameras and sensors.	Linear Regression, Polynomial regression, SVN, KNN, RF	Temperature data	63.87%
[7]	Detect and track extensive livestock systems.	GPS devices	KDE algorithm, Moving mean-based algorithm,	Collected using App-Web, GIS software tool	Graphical representation
[8]	RFID to ensure safety of beef meat.	Read and Write Microchip, IR sensor, RFID	Not applied	Local dataset	Graphical representation
[9]	Predict herd-level bovine tuberculosis	No device used	GBT	VetNet dataset	GBT 67.6% and HSp 92.3%
[10]	A thermographic system for detecting cattle diseases.	Micro-processor, Sensors, Communications.	2D Signal processing, Deep learning/Machine Learning	Privately collected data	Necessary accuracy

[11]	Real-time health monitoring.	Frequency synthesizer board	Multiple Simulation models.	Privately collected videos.	Measurement accuracy
[12]	Livestock monitoring system.	Multiple sensors.	Multiple Machine Learning, and Deep Learning models.	Collected through sensors	Multiple accuracies
[13]	System to monitor cattle health	ADXL345 accelerometer	Multiple Machine Learning models.	Privately collected data	Measurement accuracy
[14]	A consistent cow estrous cycle detection.	Devices on horns	Multiple Machine Learning algorithms	Private data on cloud database	Comparison accuracy
[15]	Get better accuracy on cattle behavioral data.	A computer workstation.	Multiple Machine Learning algorithms	Privately collected dataset.	Comparison accuracy
[16]	A Mastitis detection system using cattle behavioral data.	Portable GPS tracking device.	Multiple Machine Learning algorithms	Privately collected	Settings achieved the highest accuracy
[17]	Mastitis disease detection.	Accelerometer and magnetometer, Hub, WiFi routers.	Free learning algorithm	Cloud database systems.	90%
[18]	Mastitis disease detection.	No device used	Multiple ML and DL models	Privately collected Mastitis dataset.	J48 had 98%
[19]	Detect Clinical Mastitis in Cows	smart sac	KNN, SVM	Collected through device.	KNN 73.33% SVM 86.66%
[20]	Udder health prediction and comparison using machine learning	No device used	Multiple Machine Learning Algorithms	Privately collected data	Above 75%

Table 2.1: Summary of the Literature Review

From the above table, several systems of detecting livestock diseases can be seen. Some notable livestock diseases including Bovine Respiratory Disease, Foot and

Mouth disease, and Bovine Mastitis are detected with the use of classification algorithms and embedded devices. Live Livestock Monitoring systems with the help of devices can be seen. For the detection of diseases, several Deep Learning and Machine Learning algorithms were used such as Convolutional Neural Network, Decision Tree classification, Dynamic Time Warping, K-Nearest Neighbor, and Support Vector Machine. Several embedded devices with sensors were used to collect data and classify data using the models. Therefore, based on the analysis of several papers, it can be concluded that the use of Deep Learning algorithms and Machine Learning in conjunction with edge devices can be an effective method for the detection and diagnosis of mastitis in livestock. The high accuracy rates and real-time capabilities of these models have the potential to greatly improve the efficiency and effectiveness of disease detection in the livestock industry. Additionally, the use of embedded devices allows for continuous monitoring of animals, which can alert farmers and veterinarians to potential disease outbreaks before they become widespread. This can ultimately lead to a reduction in the spread of diseases, resulting in a positive impact on the overall health of the livestock and the economic success of the industry. So, it was concluded that the use of Deep Learning and Machine Learning algorithms in conjunction with edge devices is a promising technology for the detection and diagnosis of mastitis in livestock and can play a significant role in the advancement of the livestock industry.

2.3 Related Work and Comparision

To detect clinical mastitis in cattle, in [19], the author worked on the numerical dataset used in this paper that they colected through their device. For classification, they have used KNN and SVM algorithm and had an accuracy of 73.33% and 86.66% respectively. A comparison of several classification algorithms was done in paper [18] to get the best results for detecting mastitis. J48, Random Forest, Support Vector Machines, k-nearest Neighbor, and Naive Bayes algorithm were used on a Mastitis dataset containing 400 instances and 7 attributes taken from 100 cows on a monthly basis. All the algorithms had an accuracy of over 90% and the best performance was an accuracy of 97.75% was achieved by Random Forest and 98% achieved using the J48 algorithm. A comparision using machine learning algorithms to predict udder health was done in [20] which had an accuracy of 79.7% for both LDA and Random Forest. In [33], the study aimed to determine the effectiveness of the California Mastitis Test (CMT) in detecting intramammary infections in early lactation cows, using bacteriological culture as the gold standard. Results showed that the CMT had the highest sensitivity 82.4% and specificity 80.6%. Through our proposed model, we had an accuracy of 99.34% using InceptionV3 algorithm for Deep Learning and 99% using Random Forest for Machine Learning respectively for the detection of clinical mastitis. A comparison analysis between the traditional CMT procedure of detecting mastitis, our approach and other studies with a similar approach is displayed in the following table 2.2.

Study	Model	Accuracy
Paper [18]	Random Forest	97.75%
Paper [18]	J48	98%
Paper [19]	KNN	73.33%
Paper [19]	SVM	86.66%
Paper [20]	LDA	79.7%
Paper [20]	Random Forest	79.7%
Paper [33]	CMT	Sensitivity: 82.4% Specificity: 80.6%
Our Paper	InceptionV3 (DL)	99.34%
Our Paper	Random Forest (ML)	99%

Table 2.2: Comparision table

From the comparison of similar studies in literature, such as [18], [19] and [20], which have employed machine learning techniques for the detection of clinical mastitis, reveals that the proposed method in this paper outperforms the performance of the previously reported methods. Additionally, when compared to the traditional method of detecting clinical mastitis (CMT) presented in [33], which achieved a sensitivity of 82.4% and a specificity of 80.6%, it can be concluded that the proposed method in this paper demonstrates a significant improvement in the detection of clinical mastitis.

Chapter 3

Methodology and Dataset

In this section, we present the detailed methodology and workflow of our proposed system for detecting and diagnosing mastitis in livestock. Additionally, we provide an explanation of the dataset that we have collected and the techniques used for the visualization of the dataset.

3.1 Workflow

The total workflow of our proposed system can be divided into three sections. The dataset that has been used, consists of both image and numeric datasets. For the image data, deep learning classification is performed. For the numerical data, Machine learning is performed. Later the results of the two classifications are compared within the device to get the final result.

3.1.1 Deep Learning Workflow

In order to improve the precision of the analysis of image datasets, the images are first augmented to increase the diversity of the data. This is achieved by applying various techniques such as rotation, scaling, and flipping to the images. This allows the model to better generalize to new images. The augmented images are then divided into three sections, namely training, testing, and validation, using a split ratio of 60%, 30%, and 10%, respectively. This allows for an unbiased evaluation of the models' performance. The input data is then categorized into two classes, normal and abnormal, to facilitate the binary classification task. To achieve high precision in the classification of these image datasets, three different neural networks are implemented: ResNet50, InceptionV3, and VGG16. These models are chosen for their proven effectiveness in image classification tasks and are trained, tested and validated on the dataset. The batch size, height, and width of the images are also taken into consideration when choosing these models as it helps in making precise predictions. The models are fine-tuned to optimize their performance on the specific dataset and to improve the overall accuracy of the classification task. The approach of data augmentation and implementation of various neural network models like ResNet50, InceptionV3, and VGG16 are found to be useful in achieving high precision in image classification tasks. The fine-tuning of these models with the consideration of batch size, height, and width of the images also helps in improving the performance of the models. Figure 3.1 presents an overview of the prototype model.

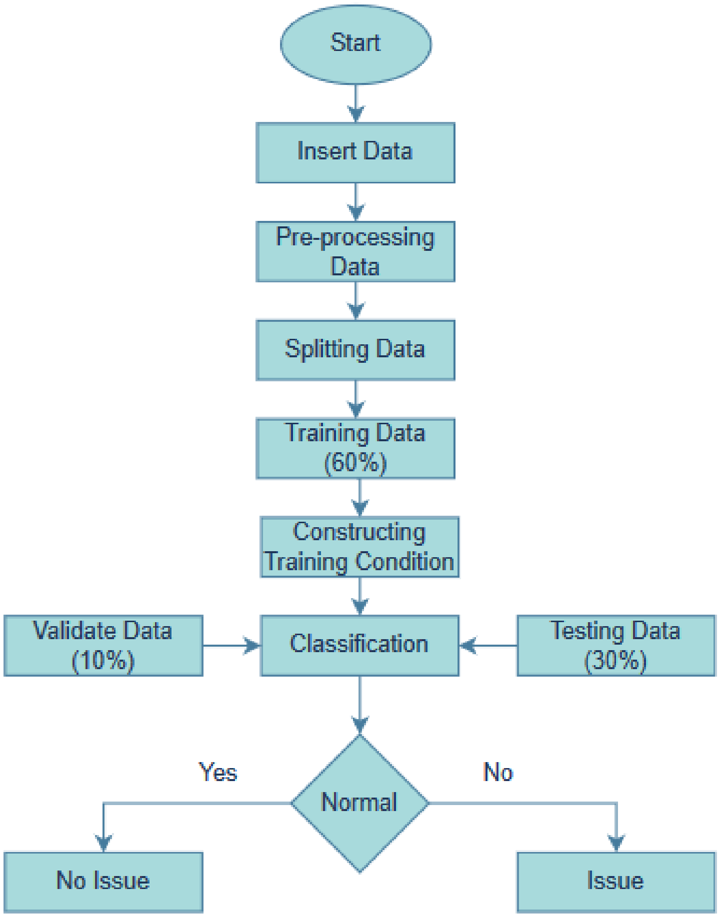


Figure 3.1: Deep Learning Workflow

3.1.2 Machine Learning Workflow

The proposed system utilizes a combination of pre-processing techniques and machine learning algorithms to classify the input data. The input data is first preprocessed using various feature engineering processes such as LabelEncoder and later the data is normalized by scaling all features on a similar scale. The data is later split into train and test in a ratio of 75% and 25% respectively. The input data is categorized into two classes 0 and 1 where 0 represents a healthy cow and 1 represents the presence of mastitis in the animal. For the classification of the data, six different machine learning models - Naive Bayes, Logistic Regression, Support VectorMachine, K-Nearest Neighbours, Decision Tree, and Random Forest was used. Out of the six models, Random Forest had the highest accuracy for the input dataset and hence it was deployed in the edge device. The following figure 3.2 presents an overview of the prototype model.

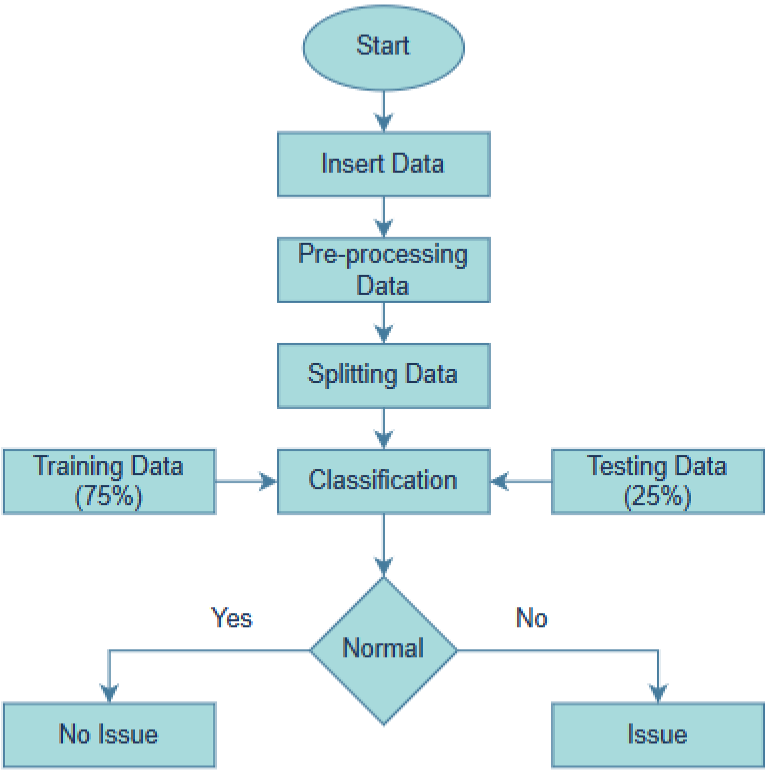


Figure 3.2: Machine Learning Workflow

3.1.3 Device Implementation Workflow

A raspberry pi connected to a pi camera and multiple sensors has been used. After the user starts the device, the pi camera is used to take pictures of the milk. The milk image is sent to the cloud service. The sensors connected to the device take in all the required data including the data that requires user input. The data that requires user input is taken from the user via a push button, after the display in the device asks for the required data input.

The collected data are then normalized and sent to the raspberry pi. Simultaneously, in the cloud, the image is classified in a virtual environment (GPU, TPU, HPC resources, etc) through a Lite model of the proposed deep learning model. The result is classified as 0 and 1 where 0 indicated normal milk and 1 indicates abnormal milk. The result is sent to the raspberry pi as Rest API. In raspberry pi, the normalized data is then sent to the Machine Learning model that has been loaded. From the Machine Learning algorithm, the result is shown as 0 or 1. Later, the result of the deep learning classification and the machine learning classification are compared by analyzing the probability of true positive and false negative of both of the results. After comparison, the final output indicating if the cow is healthy or is infected with mastitis is shown via the display connected to the device. The following figure 3.3 presents an overview of the proposed system.

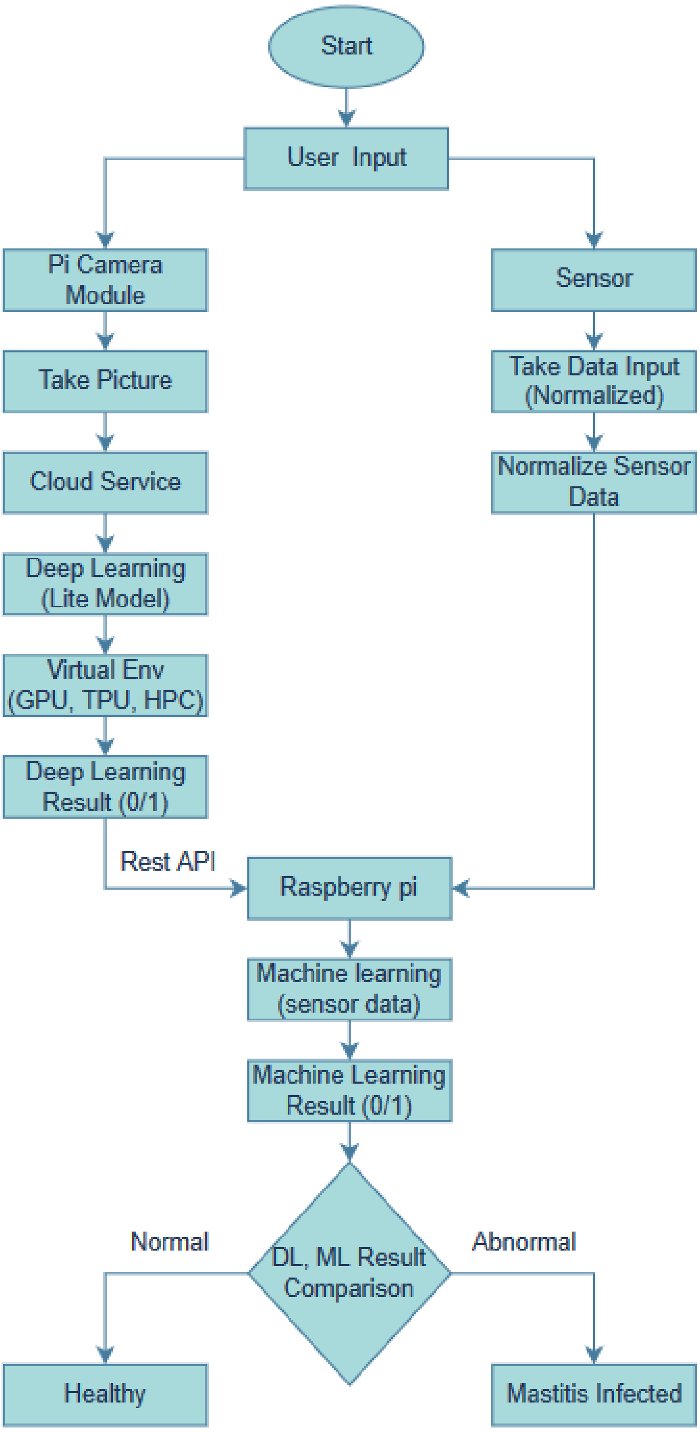


Figure 3.3: Device Implementation Workflow

3.2 Dataset

The dataset in [34] which has been used can be divided into both image and numerical data. The dataset contains the data of 1100 cows that were taken over a period of 6 days. The data is collected from a cow's udder using various sensors and techniques. Flex sensors are used to measure the size of the udder, and a temperature sensor is used to monitor the temperature of the cow. Additionally, images of the milk are processed to assess the quality of the milk, which can also be an indicator of mastitis. In the milk quality attribute, 0 indicates normal milk and 1 indicates abnormal milk. The data collected includes important information such as the cow's ID, the day and breed, the month after giving birth, past instances of mastitis, the temperature and hardness of the udder, and the size of the udder (udder front left inhale and exhale limit, udder front right inhale and exhale limit, udder rear left inhale and exhale limit, udder rear right inhale and exhale limit) as reported by a user, and pain associated with swelling. The cows are classified as either normal (0) or abnormal (1) based on the data. The figure 3.4 represents the dataset that has been used.

	Cow_ID	Day	Breed	House Number	Address	Months after giving birth	Previous_Mastits_status	IUFL	EUFL	IUFR	EUFR	IURL	EURL	IURR	EURR	Temperature	Hardness	Pain	Milk_visibility	class1
0	cow1	1	Jersey	4-1	Pudu,Bantwal TQ	1	0	150	180	150	180	150	181	150	181	43	0	0	0	0
1	cow1	2	Jersey	4-1	Pudu,Bantwal TQ	1	0	152	180	152	185	151	180	152	181	42	0	0	0	0
2	cow1	3	Jersey	4-1	Pudu,Bantwal TQ	1	0	152	182	153	186	151	186	153	183	41	0	0	0	0
3	cow1	4	Jersey	4-1	Pudu,Bantwal TQ	1	0	155	183	155	189	155	182	155	186	40	0	0	0	0
4	cow1	5	Jersey	4-1	Pudu,Bantwal TQ	1	0	150	186	150	181	150	185	150	188	41	0	0	0	0

Figure 3.4: Dataset

3.2.1 Image Data

Milk images are filtered to evaluate milk quality. The 1,341 images in this image dataset are split into two groups, "normal" and "abnormal" to help identify this condition. The images in the Normal class are those that have been unaffected by mastitis, whereas the images in the Abnormal class are those that have been infected by mastitis. These Normal and Abnormal classes are indicated as 0 and 1 in the numerical dataset under the Milk_visibility parameter.

In this study, dataset visualization techniques have been applied to this image dataset in order to better understand its characteristics. Specifically, grid layout, histogram, and principal component analysis (PCA) have been used to analyze the dataset. The grid layout has provided a comprehensive overview of the dataset, while the histogram has allowed for the examination of the distribution of pixel intensities. PCA has also helped to identify patterns and structures within the data by reducing the dimensionality of the dataset. This approach mainly has provided valuable insights into the dataset and aided in the analysis of its features and attributes.

Grid Layout:

In this study, a grid layout is used to visually inspect the data by displaying images from the bovine mastitis dataset. This layout allows us to see all of the images at once, making it easier to spot patterns or trends in the data. For example, we can compare the appearance of normal and abnormal images side by side using a grid layout, which aids in comprehending the differences between healthy and sick cows.

Figure 3.5 demonstrates the grid layout of this image dataset.

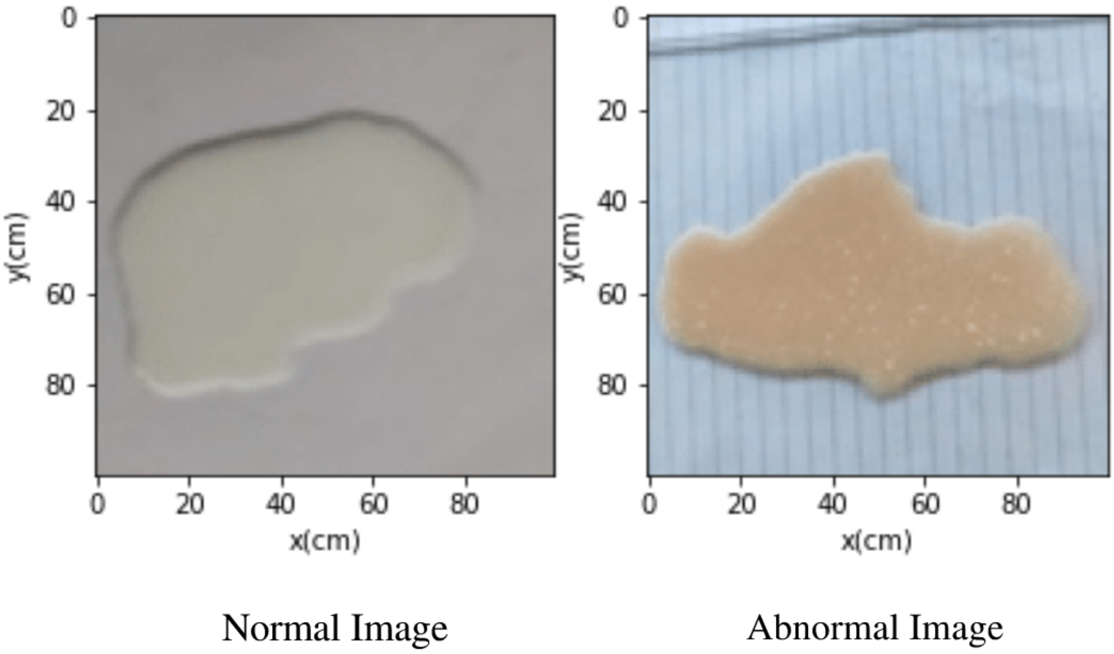


Figure 3.5: Grid Layout

Histogram:

Additionally, histograms are also applied to the dataset to quickly visualize the distribution of pixel intensities within the images. This allows us to identify patterns or trends in the data that may not be obvious when viewing the images alone. Histograms, for example, allow us to determine which pixel intensities are most common in normal images and which pixel intensities are most common in abnormal images.

Figure 3.6 demonstrates the grid layout of this image dataset.

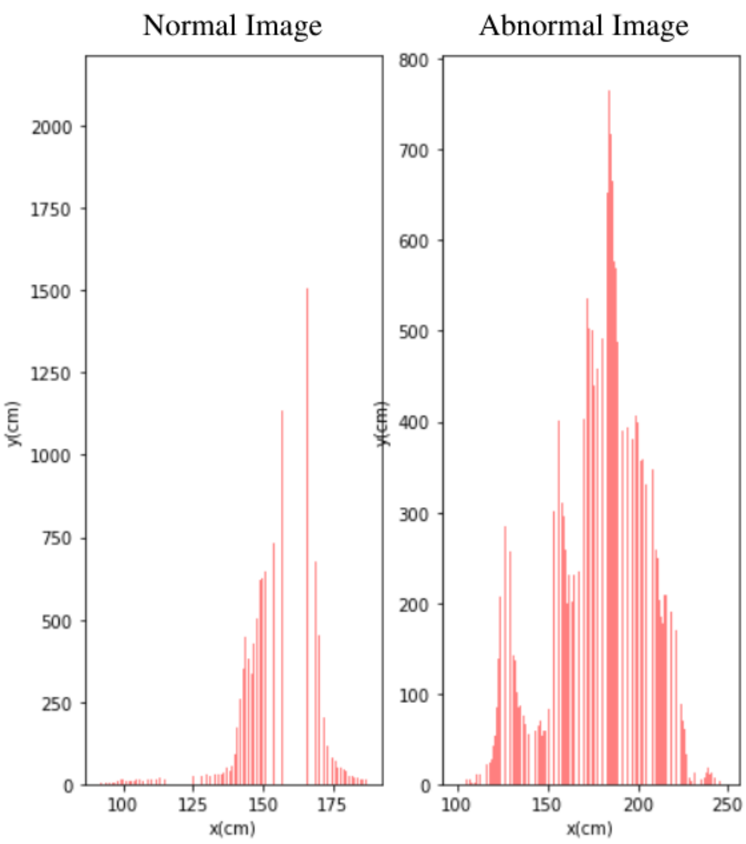


Figure 3.6: Histogram

Principal component analysis (PCA):

Principal component analysis (PCA) is applied to the dataset to understand the structure of the dataset. PCA is a technique that identifies patterns in data, by transforming the data into a new set of uncorrelated variables. This allows us to project the data onto a lower-dimensional subspace, which can make it easier to visualize patterns or trends in the data. By using this we can visualize the image dataset, where images that are similar will be plotted close to each other, while images that are dissimilar will be farther apart. The below figure 3.7 displays the principal component analysis (PCA) of the image dataset.

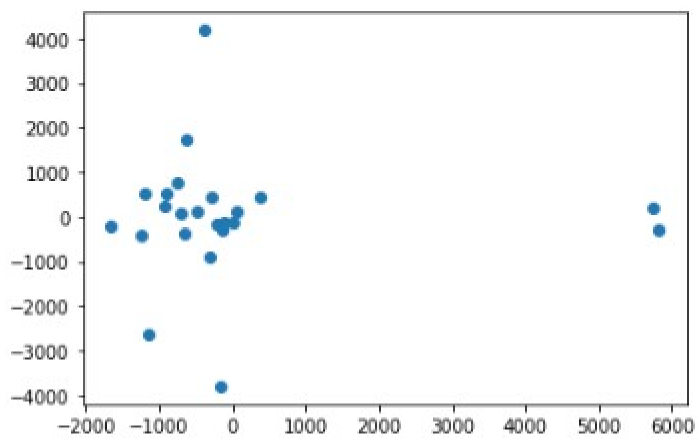


Figure 3.7: PCA

3.2.2 Numerical Data

The numerical data was collected for each cow respective to their ID over a period of six days. There are two unique breeds in the breed parameter- "Jersey" and "Holstein". The Temperature and size of the udder were taken using a temperature sensor and a flex sensor. In the dataset, the IUFL, EUFL, IUFR, EUFR, IURL, EURL, IURR, EURR represents the size of different parts of the udder of the animal which are- udder front left inhale and exhale limit, udder front right inhale and exhale limit, udder rear left inhale and exhale limit, udder rear right inhale and exhale limit respectively. The hardness and pain parameters were taken as input from the user. The Milk_visibility parameter consists of 0 and 1 which indicates Normal and Abnormal milk attributes that are obtained through image classification of the image dataset. Finally, the cows are classified as either normal (0) or abnormal (1) based on the data where the normal indicates a healthy cow and abnormal indicates a cow suffering from mastitis.

Data Analysis using Heatmap:

The following heatmap in figure 3.8 visualizes the correlation matrix of the dataset, which can be used to identify the correlation between different pairs of data.

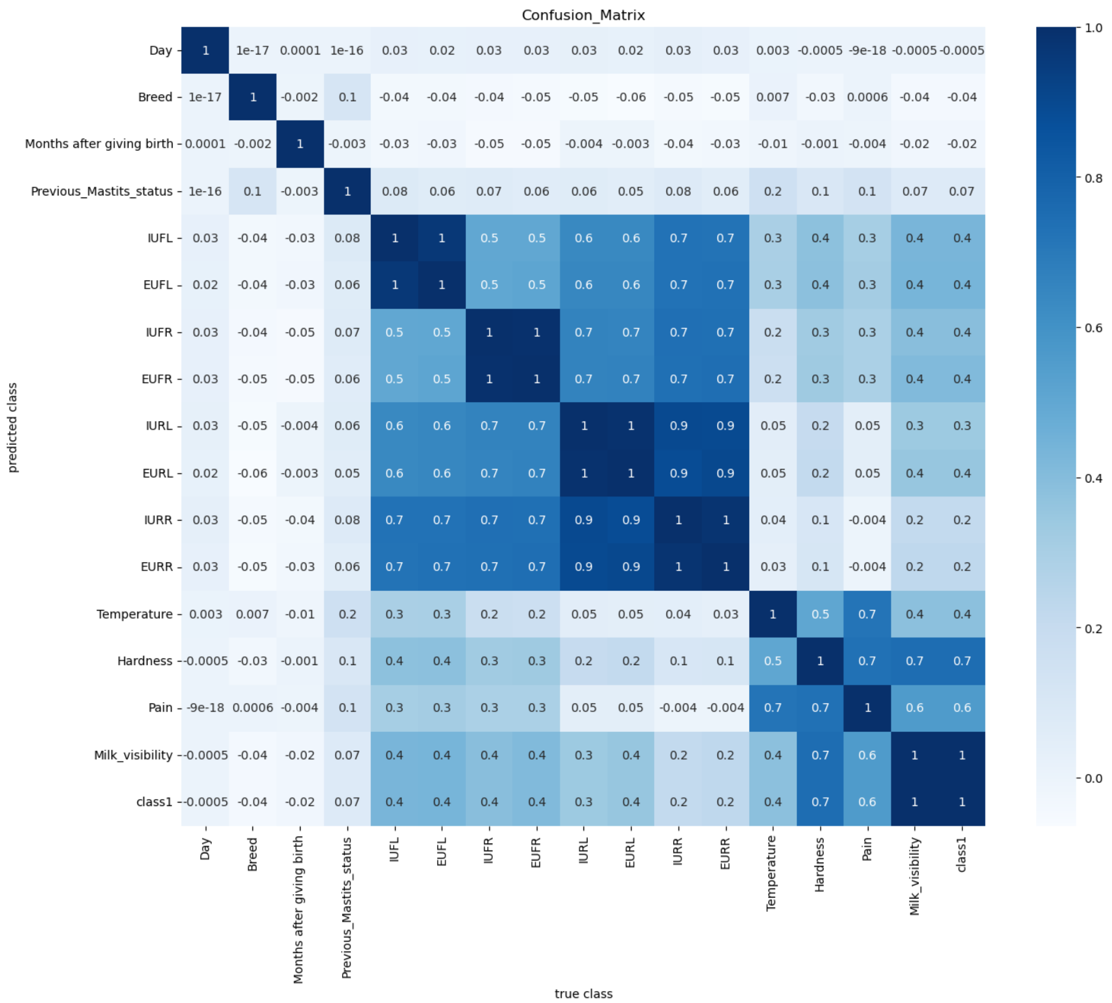


Figure 3.8: Dataset Correlation Heatmap

From the heatmap presented above, it can be observed that Milk_visibility and class1 have a perfect direct correlation, which means that as one of these parameters increases, the other parameter also increases. This indicates a strong relationship between these two parameters and may suggest that they are measuring similar or related characteristics. Additionally, it can be seen that parameters related to the temperature of the animal, the hardness of the udder, and signs of pain or distress have a high correlation with the occurrence of mastitis. This means that these parameters are positively correlated with the presence of mastitis, and may suggest that they are important factors in determining whether an animal has the condition. However, it can be noted that the type of breed, recent pregnancy of the animal, and any past cases of mastitis among the animal have very little correlation to the presence of mastitis, this indicates that these parameters are less relevant to predicting the presence of mastitis. The size of different parts of the udder also provides some correlation to the presence of mastitis. This means that the size of the udder is positively correlated with the presence of mastitis, and may suggest that it is an important factor in determining whether an animal has the condition. The day of the six-day period was found to have no correlation to the presence of mastitis in the animal. This means that there is no clear relationship between these two parameters, and may suggest that the day of the six-day period is not a relevant factor in determining whether an animal has the condition.

Chapter 4

Deep Learning Implementation and Results

This section mostly focuses on the use and anticipated outcomes of our stated model. The Keras deep learning classifiers were utilized for all implementation, testing, and validation. Three phases function has been used as the key division for the implementation purpose. In order to enhance the number of abnormal photos of the affected cows with mastitis disease, the input data was first increased. After suitable data preprocessing, input data is divided into training, testing, and validation groups. Following all of this, the ResNet50, InceptionV3, and VGG16 classification models are loaded and utilized to test and predict input data based on the amount and time required for each interpretation step. This section also exhibits the expected outcomes obtained using a confusion matrix following the application of the suggested classification models (ResNet50, InceptionV3, VGG16). The MATLAB package has been employed to visualize the findings in terms of accuracy and loss.

4.1 Data pre-processing

The $1,341$ photos in the image dataset are split into two categories—normal and abnormal—to help diagnose this condition. As a logical consequence, the presented model requires first adding new data in order to maintain the ratio for training, testing, and validation. By utilizing random oscillations and disruptions, and making sure that the class labels of the data remain the same, data augmentation encompasses a variety of techniques used to create "new" training samples from already-existing ones. The main goal of data augmentation is to increase the construct validity of the model. Adding a (few) modifications to an input image transforms its performance slightly without affecting the class name, making data augmentation a very logical and straightforward technique to use for image processing applications. Various techniques for data augmentation are frequently used to boost performance. However, we employ the "Dataset generation and expanding an existing dataset" to solve the problem we have highlighted and to provide a solution. The primary goal is to keep a reasonable ratio (train, test, validate), therefore the number of abnormal images has been placed during the augmentation process. As a response, it goes through certain stages to conclude the work. First, the disk is loaded with all of the input images. Afterward, ImageDataGenerator is applied to augment the input data. The class is initially created, and the customization for the various types of data enhancement is delivered using parameters to the class constructor. Then, numerous strategies as well as pixel scaling techniques are implemented. After reading each image by hand, it builds an array to be given to datagen using the flow method in order to handle these numerous images. Then, in order to save the modified images, it generates an iterator using the image dataset that is now in memory (using the flow() function (with parameters)). In this way, it essentially expands the training data.

4.2 Splitting Data

Once the data has been improved, it is separated into three parts for training, validation, and testing in order to guarantee precision and appropriate image measurements. The split is in the proportion of 0.6 for training, 0.1 for validation, and 0.3 for testing.

4.3 Classification

Three convolutional neural network-based models, Resnet50, InceptionV3, and VGG-16 were used to analyze a dataset on a single machine running Windows 10 Home Single Language with an AMD Ryzen 5 4600H processor, an Nvidia GeForce GTX 1650Ti graphics card, and 8GB of RAM. The performance of these models was evaluated by measuring their training and evaluation accuracy on the dataset. Table 4.1 shows the training parameters that were used for the classification.

Model Compilation	Model Optimizer = 'Adam' Loss Method ='binary-Cross Entropy'
Iteration Set	EPOCHS=100
Data Enhancement	preprocessing_function = preprocess_input, shear_range=0.2, zoom_range=0.2, horizontal_flip=True

Table 4.1: Training Parameters

4.4 Deep Learning Model Results

4.4.1 ResNet50

The resnet50 model has 98.75% accuracy after 100 epochs using input data. For that model, the confusion matrix(a) has been constructed after running all training settings. True Positive (TP) refers to the number of instances that were correctly identified as positive. In this case, 347 instances were correctly identified as having the disease. True Negative (TN) refers to the number of instances that were correctly identified as negative. In this case, 325 instances were correctly identified as not having the disease. False Positive (FP) refers to the number of instances that were incorrectly identified as positive. In this case, 9 instances were incorrectly identified as having the disease. False Negative (FN) refers to the number of instances that were incorrectly identified as negative. In this case, 1 instance was incorrectly identified as not having the disease. The heatmap(b) for that model uses colors to depict the various counts, with warmer colors (yellow) indicating higher counts and cooler colors (blue) indicating lower counts. This heatmap showcases the same data as the confusion matrix but in a more visually appealing format. This training and validation accuracy plot(c) in this research paper provides a comprehensive view of the model's performance and aids in the identification of potential issues such as overfitting or underfitting.

Figure 4.1 displays the visual representation of the heatmap and the graphical representation of the accuracy for the ResNet50 model.

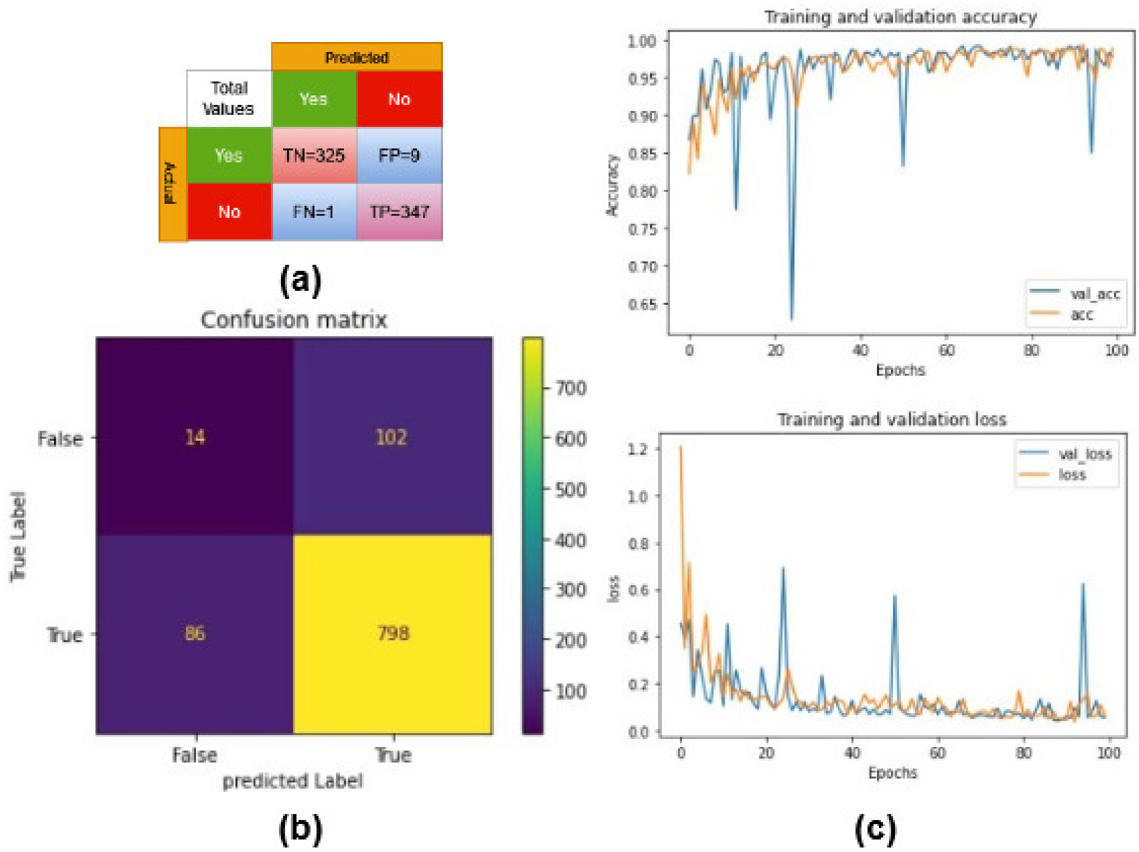


Figure 4.1: Heatmap and Accuracy plots for ResNet50

Classification report

The classification report showing the summary of this model's performance on the set of test data for which the true values are known is shown in below Table 4.2. The metrics are used and shown as precision, recall, F1-score, and support.

	Class	Precision	Recall	f1-score	Support
	Abnormal	1.00	0.97	0.98	334
	Normal	0.97	1.00	0.99	348
macro average		0.99	0.99	0.99	682
weighted average		0.99	0.99	0.99	682

Table 4.2: Classification Report for ResNet50

4.4.2 InceptionV3

For comparison analysis, following 100 epochs employing input data, the InceptionV3 model showed 99.34% accuracy. This is a positive indication that the model is learning from the training data and can generalize well to new, unknown data. For that model, the confusion matrix(a) has been constructed after running all training settings. The values in the confusion matrix represent the number of instances in each category of prediction. True Positive (TP) refers to instances that were correctly identified as positive. In this case, 346 instances were correctly identified as having the disease. True Negative (TN) refers to the instances that were correctly identified as negative. In this case, 332 instances were correctly identified as not having the disease. False Positive (FP) refers to the number of instances that were incorrectly identified as positive. In this case, 2 instances were incorrectly identified as having the disease. False Negative (FN) refers to the number of instances that were incorrectly identified as negative. In this case, 2 instances were incorrectly identified as not having the disease. For that model, the heatmap(b) uses colors to show the different counts, with warmer colors (yellow) indicating higher counts and cooler colors (blue) indicating lower counts. This heatmap shows the same information as the confusion matrix but in a more visually intuitive format. In this research paper, it can be seen that this training and validation accuracy plot(c) provides a comprehensive view of the model's performance, and helps to identify potential issues such as overfitting or underfitting.

Figure 4.2 highlights the areas of the image that the InceptionV3 model is focusing on to make its predictions. This can be useful for understanding where the model is making its decisions and identifying any potential areas of improvement. Additionally, the graphical representation of accuracy in Figure 4.2 allows for easy comparison of the model's performance.

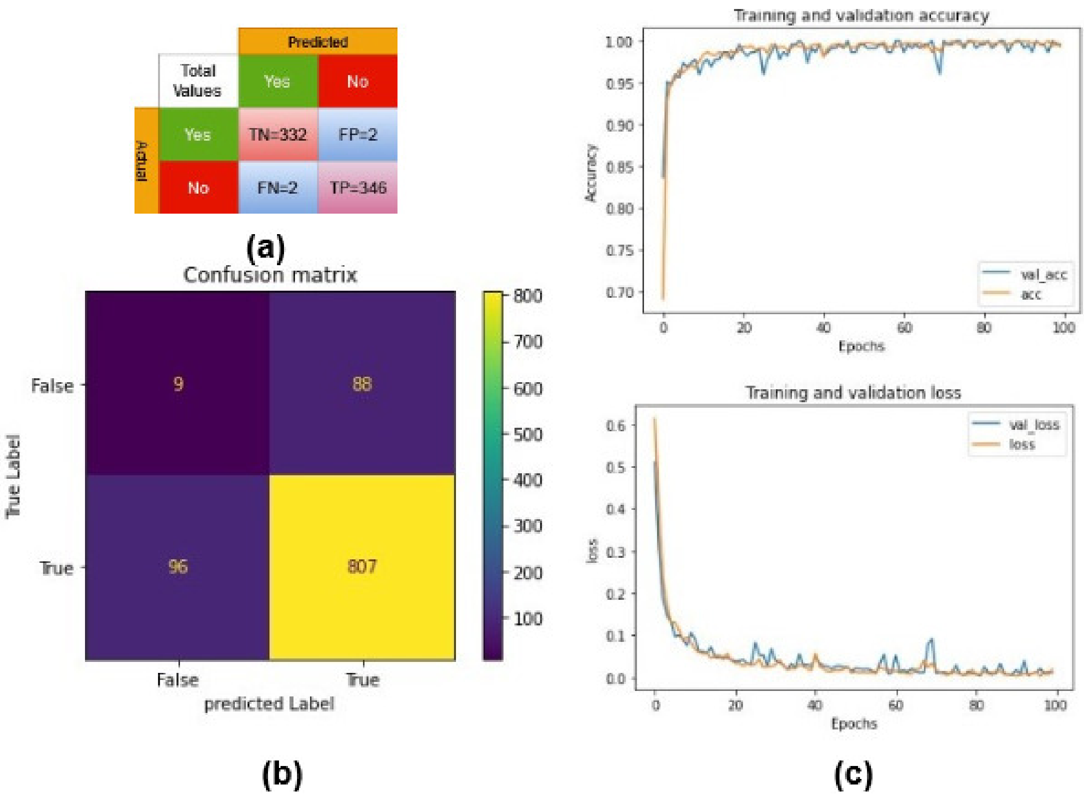


Figure 4.2: Heatmap and Accuracy plots for InceptionV3

Classification report

The classification report showing the summary of this model's performance on the set of test data for which the true values are known is shown in below Table 4.3. The metrics are used and shown as precision, recall, F1-score, and support.

	Class	Precision	Recall	f1-score	Support
	Abnormal	0.99	0.99	0.99	334
	Normal	0.99	0.99	0.99	348
macro average		0.99	0.99	0.99	682
weighted average		0.99	0.99	0.99	682

Table 4.3: Classification Report for InceptionV3

4.4.3 VGG16

After implementation, the VGG16 model achieved 98.01% accuracy over 100 epochs utilizing the input dataset.This is a promising indicator that the model is learning from the training instances and can make generalizations well to new, previously unknown information. After operating all training configurations, the confusion matrix(a) for that model has been constructed. The numbers in the confusion matrix represent the number of instances in each prediction category. The instances that were correctly identified as positive are referred to as the True Positive (TP).In this case, 348 instances of the disease were correctly identified. The instances that were correctly identified as negative are referred to as the True Negative (TN).In this case, 299 instances were correctly identified as being free of the disease. The number of instances that were incorrectly identified as positive is referred to as the False Positive (FP).In this case, 35 instances were mistakenly identified as having the disease. The number of instances that were incorrectly identified as negative is referred to as the False Negative (FN).In this case, 0 instances were incorrectly identified as lacking the disease. For that model, The heatmap(b) for the model uses colors to show the different counts, with warmer colors like yellow indicating high counts and cooler colors like blue indicating lower counts. This heatmap presents the same information as the confusion matrix but in a more visually intuitive format that makes it easier to understand the model's performance. The training and validation accuracy plot (c) in this research paper gives a complete understanding of the model's performance and allows for the identification of any potential issues, like overfitting or underfitting.

Figure 4.3 displays the visual representation of the heatmap and the graphical representation of the accuracy for the VGG16 model.

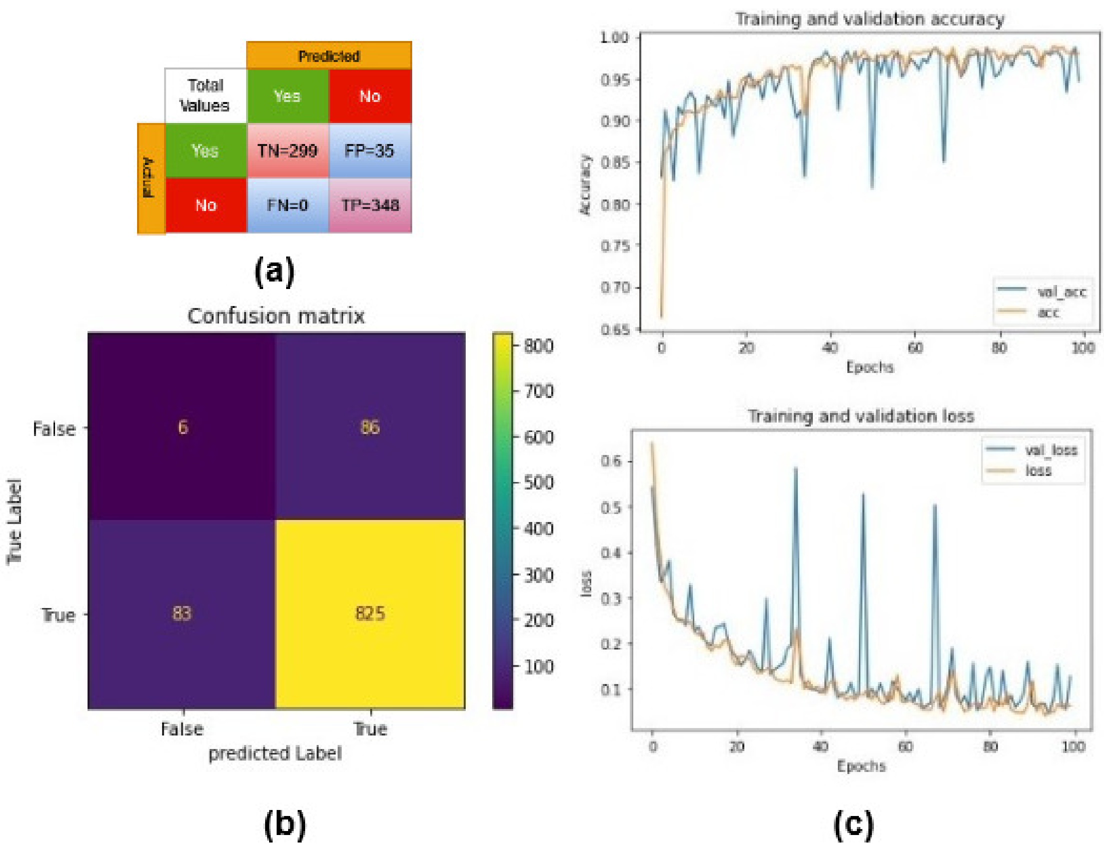


Figure 4.3: Heatmap and Accuracy plots for VGG16

Classification report

The classification report showing the summary of this model's performance on the set of test data for which the true values are known is shown in below Table 4.4. The metrics are used and shown as precision, recall, F1-score, and support.

	Class	Precision	Recall	f1-score	Support
	Abnormal	1.00	0.90	0.94	334
	Normal	0.91	1.00	0.95	348
macro average		0.95	0.95	0.95	682
weighted average		0.95	0.95	0.95	682

Table 4.4: Classification Report for VGG16

4.5 Comparison and Analysis

The InceptionV3 model performed the best with an accuracy of 99.34% after 100 epochs using the input dataset. The VGG16 model had an accuracy of 98.01% and the ResNet50 model had an accuracy of 98.75%. In terms of comparison, the InceptionV3 model had a higher accuracy than both the VGG16 and ResNet50 models, indicating that it is the best-performing model out of the three. The InceptionV3 model was a particularly strong and suitable choice for this task due to its state-of-the-art performance on image classification benchmarks and its ability to handle a wide range of image sizes and scales from the other two models(ResNet50, VGG16). The InceptionV3 model is a widely-used image classification model that has achieved impressive results on a number of image classification tasks. Its modular design, which involves a combination of convolutional and pooling layers for extracting low-level features and fully-connected layers for learning higher-level features, allows the model to learn complex patterns in the data and make more accurate predictions. Additionally, the InceptionV3 model is able to effectively classify images despite variations in orientation, lighting, and other factors that can affect image quality, making it well-suited for the Bovine Mastitis disease detection task.

In summary, our approach of using image augmentation and state-of-the-art image classification models was highly effective for detecting Bovine Mastitis disease in cows. The InceptionV3 model performed particularly well on this task, achieving high accuracy and good performance in the accuracy graph and classification report. This suggests that the model is capable of effectively learning from the input data and making accurate predictions about the health status of cows based on images of their udders. The use of image augmentation techniques such as fine-tuning and different activation functions, as well as the modular design of the InceptionV3 model, likely contributed to the model's strong performance on this task. Results obtained from all the algorithms are shown in table 4.5 and figure 4.4 below.

Classification Algorithm	Accuracy
ResNet-50	98.75%
InceptionV3	99.34%
VGG16	98.01%

Table 4.5: Model Accuracy Comparison Table

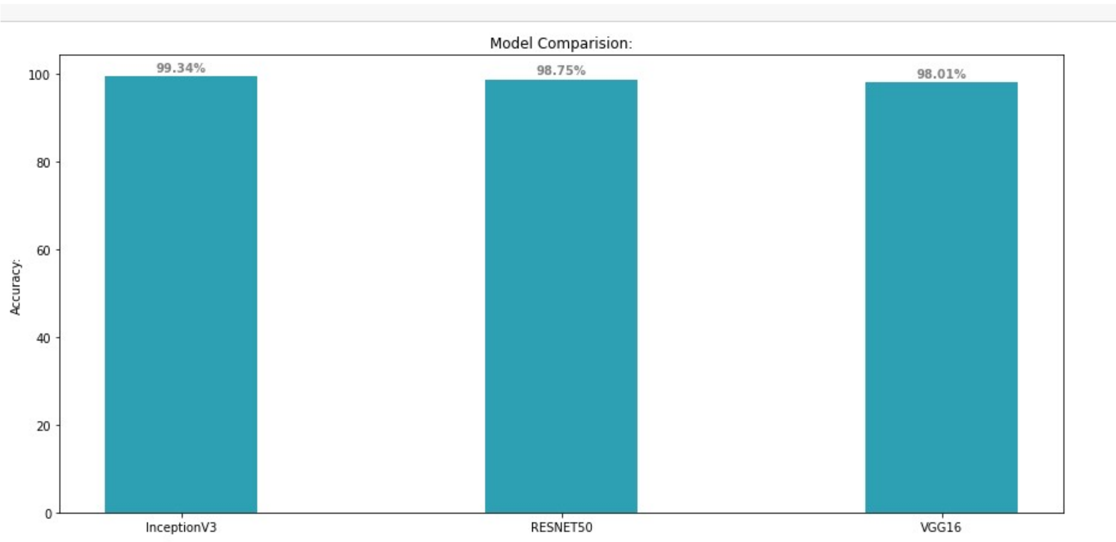


Figure 4.4: Comparison of Model Accuracies

In this study, we have utilized the Netron visualization tool to analyze the internal structure of the InceptionV3 model. Through this analysis, we have gained a deeper understanding of the model's functionality and performance. The visualization revealed the presence of several key layers within the model, including convolutional, max pooling, batch normalization, and activation layers. These layers are the building blocks of the InceptionV3 model and play a crucial role in its performance. The convolutional layers are responsible for extracting features from the input images. These features are then passed through the max pooling layers, which reduce the spatial resolution of the feature maps and help to reduce the computational complexity of the model. The batch normalization layers are used to normalize the activations of the previous layers, which helps to improve the stability of the model during training. Finally, the activation layers are used to introduce non-linearity into the model, allowing it to learn more complex representations of the input data.

Overall, our analysis of the InceptionV3 model using the Netron visualization tool has offered valuable insights into the model's functionality and performance. By gaining a deeper understanding of the model's architecture, we can improve its performance and apply it to a wider range of tasks. Moreover, our analysis also highlights the importance of proper layer design and architecture choices in achieving optimal performance. By comparing the performance of the InceptionV3 model to other architectures, we can gain insight into the trade-offs between different design choices, such as the number of layers, the filter sizes, and the number of filters. Figure 4.5 below demonstrates the internal functionality of InceptionV3 model.

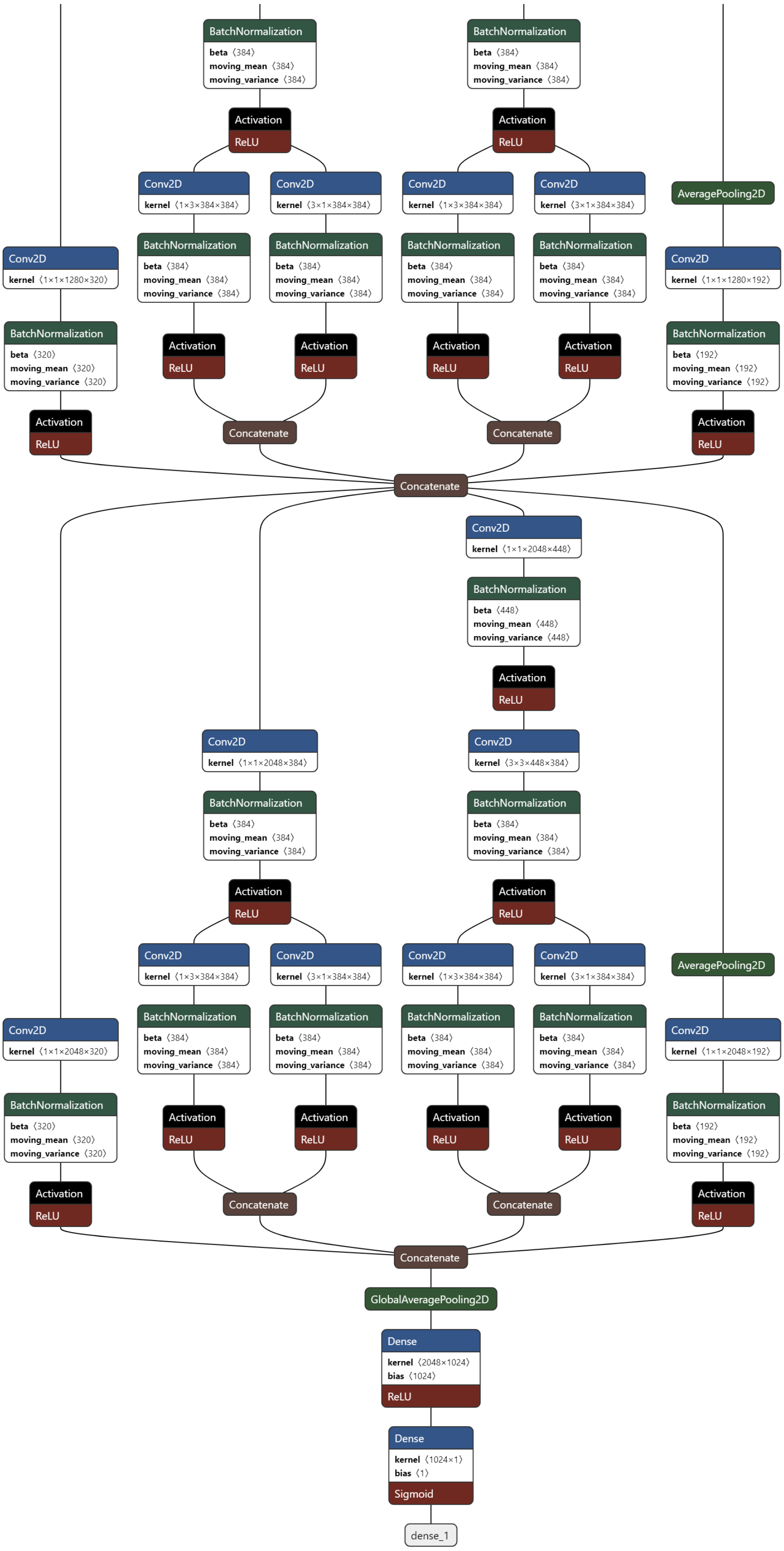


Figure 4.5: Internal Architecture Visualization of this purpose model

Chapter 5

Machine Learning Implementation and Results

For the numerical dataset, a machine learning comparison analysis was done in which the dataset was first pre-processed through feature engineering, and then split into training and testing sets. Six different Machine learning models were trained on the training set and evaluated on the test set. The models used were Random Forest, K-nearest Neighbors, Support vector machine, Decision tree, Logistic Regression, and naive Bayes. The results of the evaluation indicate that the models performed with varying degrees of accuracy, with Random Forest, Decision Tree, and K-nearest Neighbors achieved the highest accuracy at 99%, 97%, and 96%, respectively. The support vector machine and Logistic Regression models performed well with an accuracy of 93%. The Naive Bayes model had the lowest accuracy at 92%.

5.1 Data Preprocessing

Since the Milk_visibility parameter had a direct correlation with the class1, the parameter Milk_visibility could not be used as a parameter for the classification in order to detect the final class1. Moreover, the Cow-ID, Day, House Number, and Address parameters had to be dropped from the dataset as they had no correlation with the occurrence of mastitis. The “Breed” parameter had two unique values- “Jersey” and “Holstein”. This values were converted to 0 and 1 with LabelEncoder. With 0 representing “Jersey” and 1 representing “Holstein”.

LabelEncoder works by assigning an integer value to each unique category in the column. It first finds all the unique categories in the column and then assigns an integer value to each of them, starting from 0. Then, it maps each category in the column to the corresponding integer value. The code creates an instance of the LabelEncoder class and assigns it to a variable. Then it uses this instance to convert the ”Breed” column of the data DataFrame to numerical values by calling the fit_transform() method on the ”Breed” column and assigns the result back to the 'Breed' column. The result is a new DataFrame with the 'Breed' column containing numerical values. The following figure 5.1 shows the dataset, after the data preprocessing through feature engineering.

	Breed	Months after giving birth	Previous_Mastits_status	IUFL	EUFL	IUFR	EUFR	IURL	EURL	IURR	EURR	Temperature	Hardness	Pain	class1
0	0	1	0	150	180	150	180	150	181	150	181	43	0	0	0
1	0	1	0	152	180	152	185	151	180	152	181	42	0	0	0
2	0	1	0	152	182	153	186	151	186	153	183	41	0	0	0
3	0	1	0	155	183	155	189	155	182	155	186	40	0	0	0
4	0	1	0	150	186	150	181	150	185	150	188	41	0	0	0

Figure 5.1: Preprocessed Dataset

Later, using StandardScaler the features in the dataset was normalized. Through Normalization or feature scaling, the data within the features of the dataset were scaled to ensure that all features are on a similar scale, which is important for the machine learning algorithms to work correctly.

5.2 Splitting Data

After the feature engineering and normalization of the dataset, a train-test split is performed on the dataset. The dataset is split into training and testing sets. The test_size parameter is set to 0.25, meaning that 25% of the data will be used for testing and 75% for training. The random_state parameter is set to 42, which is used as a seed for the random number generator that is used to select the data points for the training and testing sets, so the same set of data will be chosen every time the program runs with the same seed.

5.3 Hyperparameter Tuning

Using GridSearch, the hyperparameters of the machine learning models were tuned, and the best possible set of parameters was selected for the application of the machine learning models. GridSearch is a class that performs an exhaustive search over a specified parameter space and cross-validates the model using different combinations of the parameters to find the best set of hyperparameters to optimize the model performance.

5.4 Machine learning models and results

For the classification, six different Machine Learning models were used- Random Forest, Decision Tree, K-Nearest Neighbors, Support Vector Machine, Logistic regression, and Naive Bayes. For each of the model, the heatmap of the confusion matrix, the learning curve, and the classification report was generated. The learning curve is a graphical representation of the relationship between the model's performance which is measured as a score and the size of the training set. The curve shows how the model's performance changes as more data are used for training. The accuracy, heatmap of the confusion matrix, the learning curve, and the classification report of each of the model that has been used for the classification of the dataset has been described below.

5.4.1 Random Forest

Random Forest is a supervised learning algorithm which is an ensemble method that combines the predictions of multiple models or decision trees to make a final prediction. Implementing Random Forets for the given dataset had an accuracy of 99%.

The heat map of the confusion matrix and the learning curve for the Random forest algorithm is shown in Fig 5.2. In the figure, (a) represents the heatmap of the confusion matrix and (b) represents the learning curve.

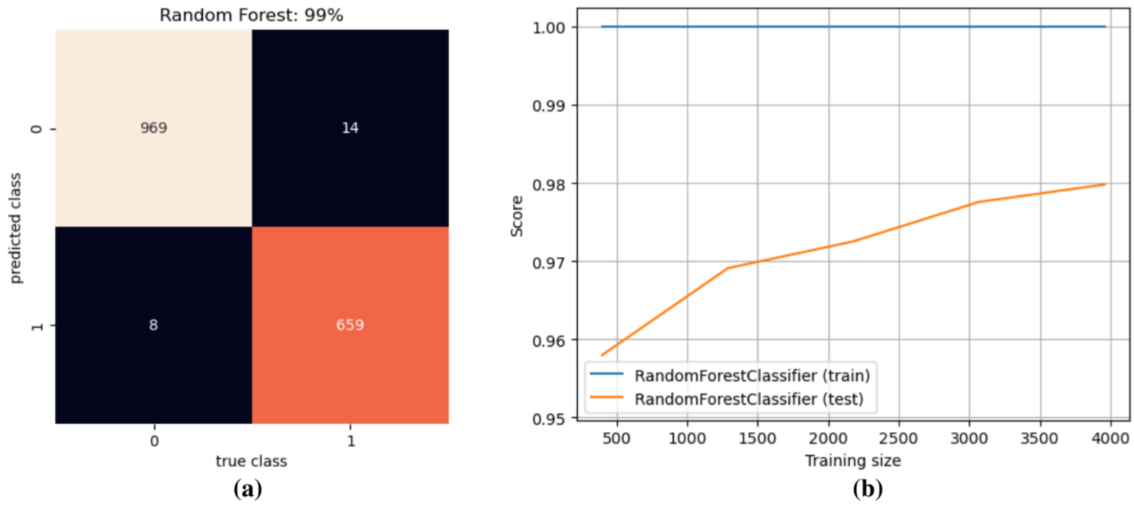


Figure 5.2: Heatmap and Learning Curve for Random Forest

Classification report

The following table 5.1 summarizes the performance of the Random Forest algorithm through the classification report. Here, Each row represents a class where the possible classes are 0 and 1, and the columns show the precision, recall, F1-score, and support.

	Class	Precision	Recall	f1-score	Support
	0	0.99	0.99	0.99	977
	1	0.99	0.98	0.99	673
macro average		0.99	0.99	0.99	1650
weighted average		0.99	0.99	0.99	1650

Table 5.1: Classification Report for Random Forest algorithm

5.4.2 Decision Tree

Decision Tree is a supervised learning algorithm that uses a tree-like model of decisions to predict outcomes. It splits the data recursively based on the values of the input features and each leaf node represents a class label or a predicted value. Implementing Decision Tree algorithm for the given dataset had an accuracy of 97%.

The heat map of the confusion matrix and the learning curve for the Decision Tree algorithm is shown in Fig 5.3. In the figure, (a) represents the heatmap of the confusion matrix and (b) represents the learning curve.

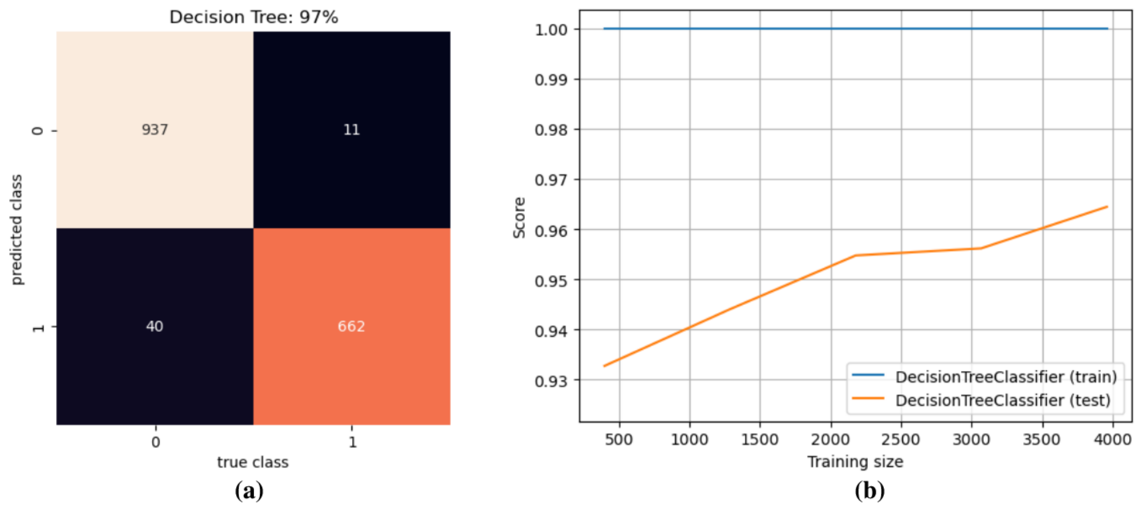


Figure 5.3: Heatmap and Learning Curve for Decision Tree

Classification report

The following table 5.2 summarizes the performance of the Decision Tree algorithm through the classification report. Here, Each row represents a class where the possible classes are 0 and 1, and the columns show the precision, recall, F1-score, and support.

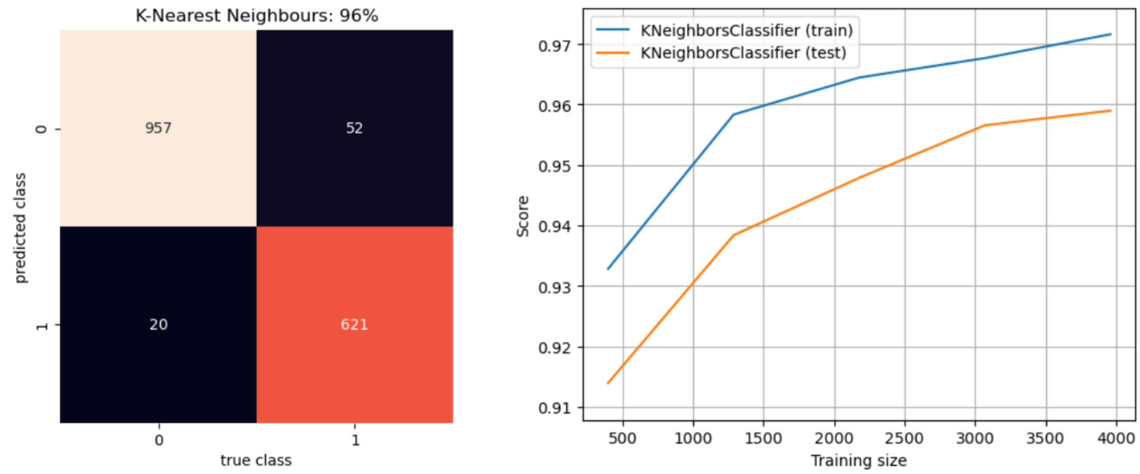
	Class	Precision	Recall	f1-score	Support
	0	0.99	0.96	0.97	977
	1	0.94	0.98	0.96	673
macro average		0.96	0.97	0.97	1650
weighted average		0.97	0.97	0.97	1650

Table 5.2: Classification Report for Decision Tree

5.4.3 K-Nearest Neighbors (KNN)

K-Nearest Neighbors is a supervised learning algorithm that works by finding the k-number of nearest points to a new data point and using the majority class or average value among those points to make a prediction. Implementing K-Nearest Neighbors for the given dataset had an accuracy of 96%.

The heat map of the confusion matrix and the learning curve for the K-Nearest Neighbors algorithm is shown in Fig 5.4. In the figure, (a) represents the heatmap of the confusion matrix and (b) represents the learning curve.



(a) **(b)**

Figure 5.4: Heatmap and Learning Curve for K-Nearest Neighbors (KNN)

Classification report

The following table 5.3 summarizes the performance of the K-Nearest Neighbors algorithm through the classification report. Here, Each row represents a class where the possible classes are 0 and 1, and the columns show the precision, recall, F1-score, and support.

	Class	Precision	Recall	f1-score	Support
	0	0.95	0.98	0.96	977
	1	0.97	0.92	0.95	673
macro average		0.96	0.95	0.95	1650
weighted average		0.96	0.96	0.96	1650

Table 5.3: Classification Report for K-Nearest Neighbors algorithm

5.4.4 Support Vector Machine (SVM)

Support Vector Machine is a type of supervised learning algorithm that aims to find the optimal boundary or hyperplane that separates data into different classes or predicts the target value, by maximizing the margin between the different classes. Implementing Support Vector Machine for the given dataset had an accuracy of 93%.

The heat map of the confusion matrix and the learning curve for the Support Vector Machine algorithm is shown in Fig 5.5. In the figure, (a) represents the heatmap of the confusion matrix and (b) represents the learning curve.

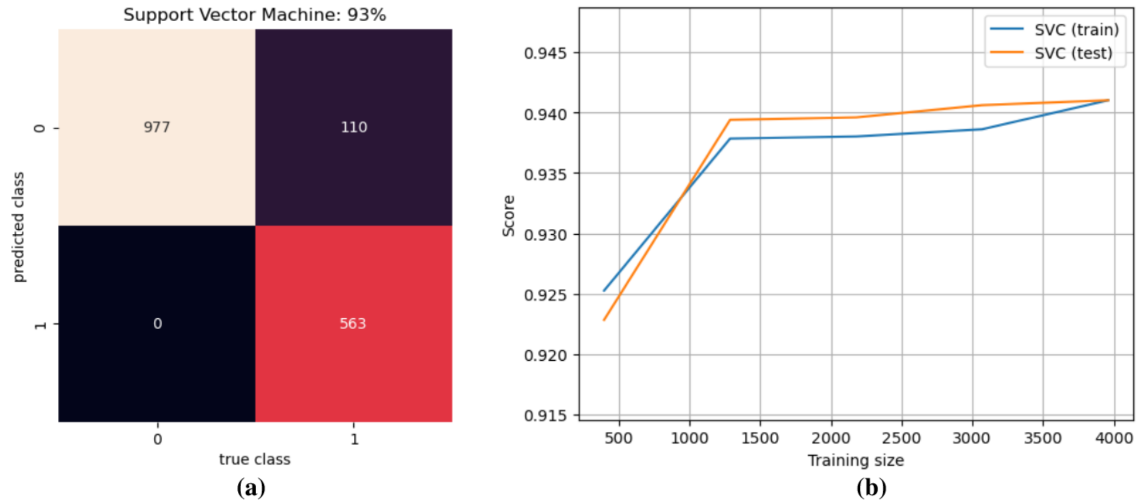


Figure 5.5: Heatmap and Learning Curve for Support Vector Machine (SVM)

Classification report

The following table 5.4 summarizes the performance of the Support Vector Machine algorithm through the classification report. Here, Each row represents a class where the possible classes are 0 and 1, and the columns show the precision, recall, F1-score, and support.

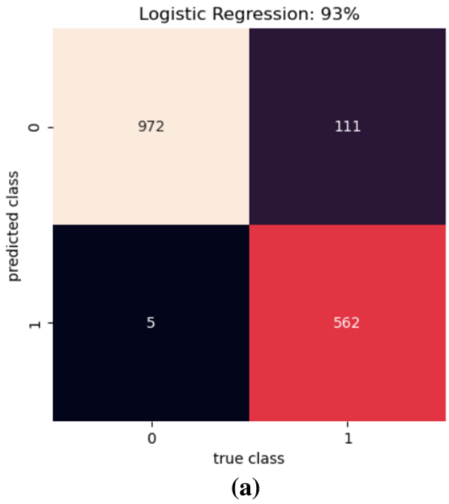
	Class	Precision	Recall	f1-score	Support
	0	0.90	1.00	0.95	977
	1	1.00	0.84	0.91	673
macro average		0.95	0.92	0.93	1650
weighted average		0.94	0.93	0.93	1650

Table 5.4: Classification Report for Support Vector Machine algorithm

5.4.5 Logistic regression

Logistic Regression is a supervised learning algorithm that models the probability of a certain class or event occurring given the values of the input features. It finds the best coefficients of the input features by minimizing the difference between the predicted probability and the true labels, using a logistic function. Implementing Logistic regression for the given dataset had an accuracy of 93%.

The heat map of the confusion matrix and the learning curve for the Logistic regression Machine algorithm is shown in Fig 5.6. In the figure, (a) represents the heatmap of the confusion matrix and (b) represents the learning curve.



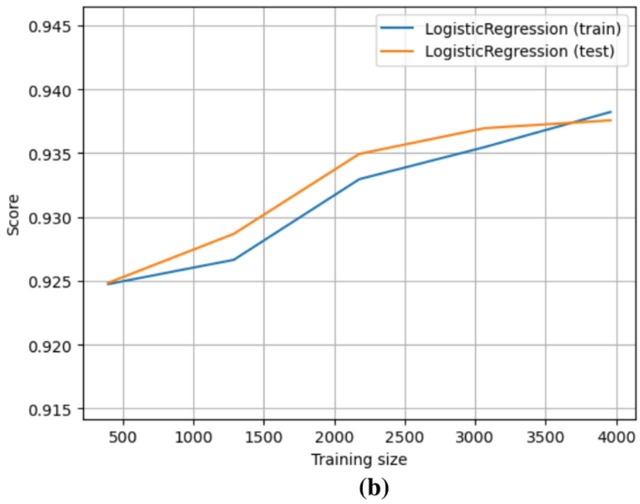


Figure 5.6: Heatmap and Learning Curve for Logistic regression

Classification report

The following table 5.5 summarizes the performance of the Logistic regression algorithm through the classification report. Here, Each row represents a class where the possible classes are 0 and 1, and the columns show the precision, recall, F1-score, and support.

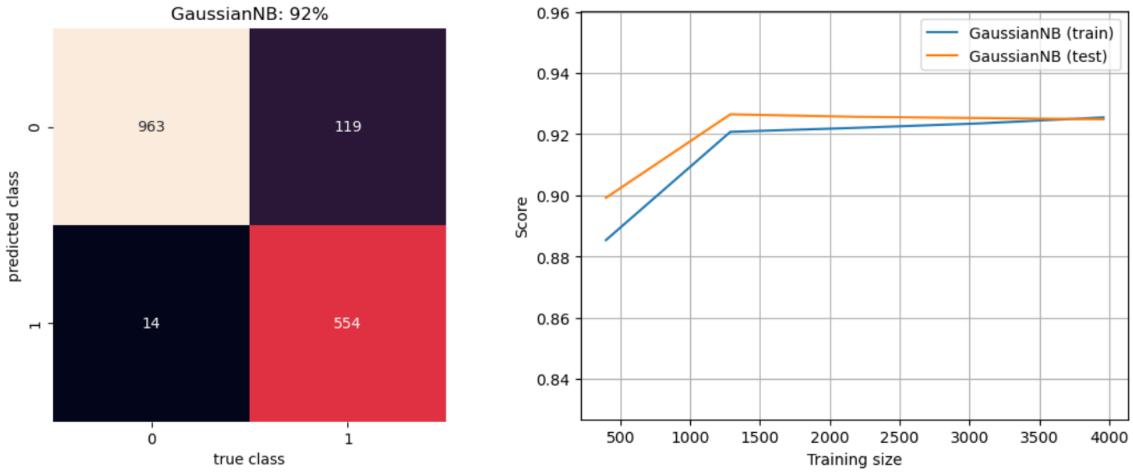
	Class	Precision	Recall	f1-score	Support
	0	0.90	0.99	0.94	977
	1	0.99	0.84	0.91	673
macro average		0.94	0.91	0.93	1650
weighted average		0.94	0.93	0.93	1650

Table 5.5: Classification Report for Logistic regression algorithm

5.4.6 Naive Bayes

Naive Bayes is a probabilistic algorithm that uses Bayes' theorem to classify data based on the probability of a certain class or event occurring given the values of the input features. Implementing Naive Bayes for the given dataset had an accuracy of 92%.

The heat map of the confusion matrix and the learning curve for the Naive Bayes Machine algorithm is shown in Fig 5.7. In the figure, (a) represents the heatmap of the confusion matrix and (b) represents the learning curve.



(a) (b)

Figure 5.7: Heatmap and Learning Curve for Naive Bayes

Classification report

The following table 5.6 summarizes the performance of the Naive Bayes algorithm through the classification report. Here, Each row represents a class where the possible classes are 0 and 1, and the columns show the precision, recall, F1-score, and support.

	Class	Precision	Recall	f1-score	Support
	0	0.89	0.99	0.94	977
	1	0.98	0.82	0.89	673
macro average		0.93	0.90	0.91	1650
weighted average		0.92	0.92	0.92	1650

Table 5.6: Classification Report for Naive Bayes algorithm

5.5 Comparison and Analysis

From the individual classification of all the models, it is noted that Random Forest performed the best with an accuracy of 99%. The Decision Tree and K-Nearest Neighbors models also performed well with an accuracy of 97% and 96% respectively. Logistic Regression, Support Vector Machine and Naive Bayes models performed relatively lower with an accuracy of 93%, 93%, and 92% respectively.

From all of the models that were used, Random Forest and Decision Tree models performed the best among all other models and are suitable for the problem at hand. K-Nearest Neighbors also performed well and could be considered as an alternative model. The Logistic Regression, Support Vector Machine and Naive Bayes models performed relatively lower and may not be the best choice for the problem at hand. The comparison between the models is visualized in figure 5.8 below.

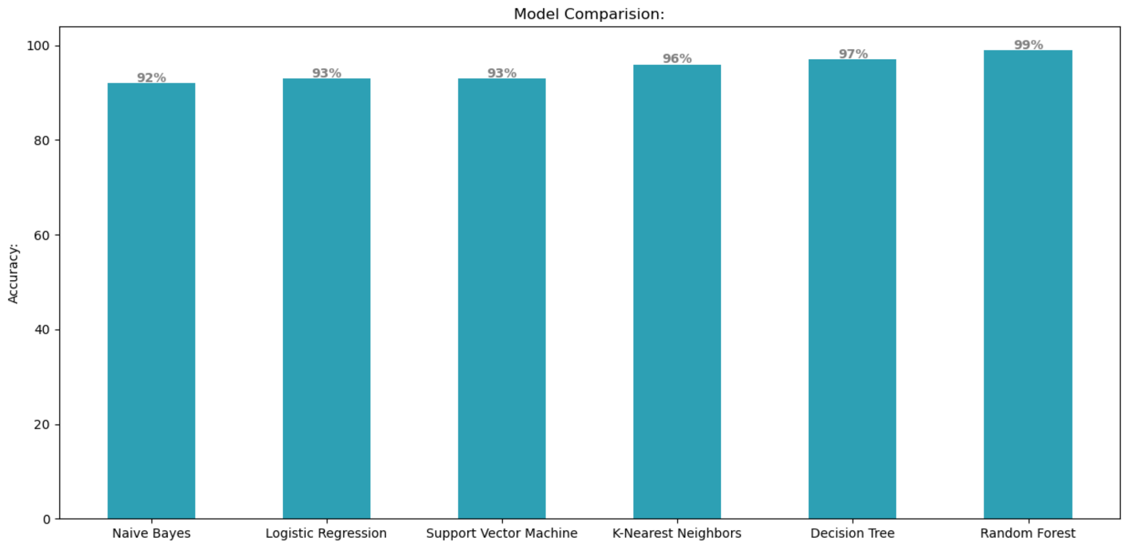


Figure 5.8: Machine Learning Accuracy Comparison

5.6 Calibration Curve

The following figure 5.9 represents the calibration curve for all of the modes that were used. The calibration curve is a plot that compares the predicted probabilities of a model to the actual outcomes. The x-axis of the plot represents the predicted probability of the model and the y-axis represents the true positive rate. The diagonal line represents the ideal case where the predicted probabilities perfectly match the true outcomes. The closer the plotted points are to this line, the better calibrated the model is. The farther away from the line, the more poorly calibrated the model is. Therefore, from the curve, it is seen that the Random forest algorithm is performing the best as it is closest to the ideal line and Naive Bayes is performing the least as it is farthest from the ideal line.

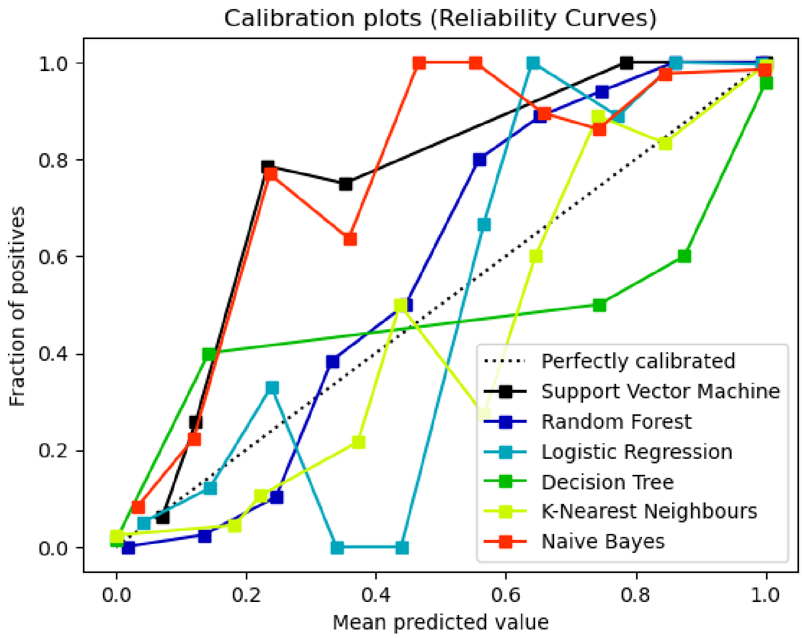


Figure 5.9: Calibration Curve

Chapter 6

Device Deployment

6.1 Proposed Architecture of the Device

The architecture of the proposed device is shown in Figure 6.1. The device is used to take in data, classify the models and display the result.

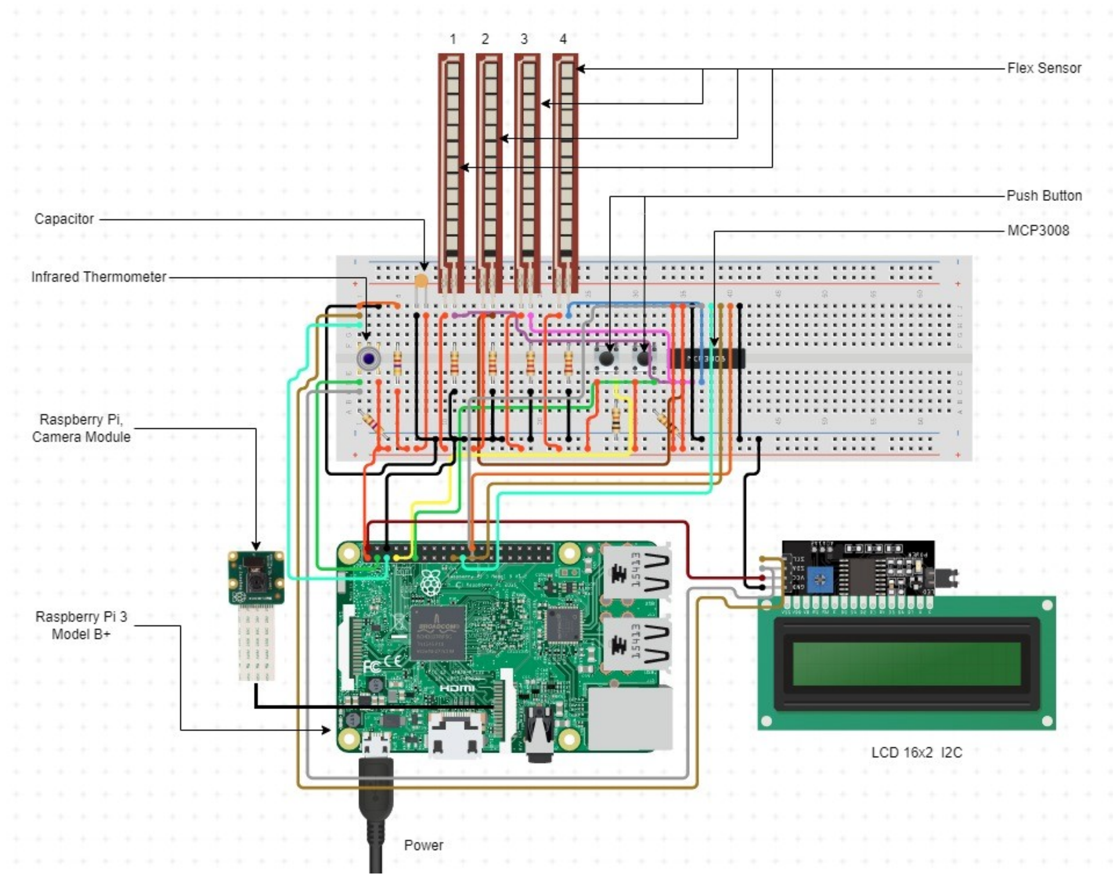


Figure 6.1: Circuit Diagram

The proposed device uses a Raspberry Pi 3. With the Raspberry pi, a pi camera, four flex sensors, an infrared thermometer two push buttons, one LCD 16×2 I2C display, and an MCP3008 has been used. The pi camera is used to take pictures of the milk of the cow and store them in the raspberry pi or any other cloud service.

The four flex sensors are used to take in the data of the four parts of the udder of the cow. The flex sensors when bent or flexed gives a resistance value proportional to the fixation. By calibrating the values of the resistance received from the flex sensors, the shape of the four parts of the cow's udder is determined. The MCP3008 is used for the conversion of analog input signals received by the flex sensors into digital values, which can be read and processed by the raspberry pi. An infrared thermometer is connected to collect the temperature of the cow. A capacitor ceramic 100nF is used to filter out the noise and stabilize the voltage while using the infrared thermometer. Two push buttons are connected to the device to get user inputs. For the "Pain" and "Hardness" parameters of the dataset, the user has to select between "Yes" or "No" input, and for the"Breed Parameters," the user has to select between "Jersey" or "Holstein" which can be done through the two push buttons. Moreover, one of the push buttons is used to take pictures of the milk. The user will be asked for inputs through the 16×2 LCD display. The display will also be used to display the results. The whole device can be powered both through a wired connection or wirelessly with a power source.

The user first takes a picture of the image, which is subsequently transmitted to a cloud service. Then this deep learning model (inceptionV3) is deployed on a virtual machine within the cloud service, and the resulting output is stored in a model registry. The output is then transformed into a REST API, which is subsequently sent back to the device. Simultaneously, after the image is taken, the device will ask to proceed toward sensor inputs. The sensor data will be calibrated and sent to the raspberry pi. In the raspberry pi, the sensor data is pre-processed and normalized. The set of collected data is saved as an array and sent to the model predict function. In the model predict function, the Machine Learning Model which was saved and loaded in the raspberry pi as a pickle file takes in the data array and runs the classification. The result is saved as 0 or 1 where 0 represents normal and 1 represents abnormal The model selected for the system is Random Forest which had the highest accuracy at 99%. The result of deep learning and machine learning is later compared by analyzing the probability of true positive and false negative of both of the results. After comparison, the final output indicating if the cow is healthy or is infected with mastitis is shown through the LCD display.

6.2 Hardware Limitations

Because of the unavailability of the sensors and real-world data, the Machine Learning model was implemented on the raspberry pi with the test data from the dataset which was taken as input from the user through a keyboard connected to the raspberry pi. The below Figure 6.2 represents the currently available hardware of the device, which contains a raspberry pi 3 and a raspberry pi camera module. Figure 6.3 represents the raspberry pi terminal where input data is displayed as an array and the corresponding predicted result for this data is shown in the predicted result as an array of either [0] or [1]. Here, [0] in the predicted result represents a healthy cow, which is correct and corresponds to the data taken from the dataset.

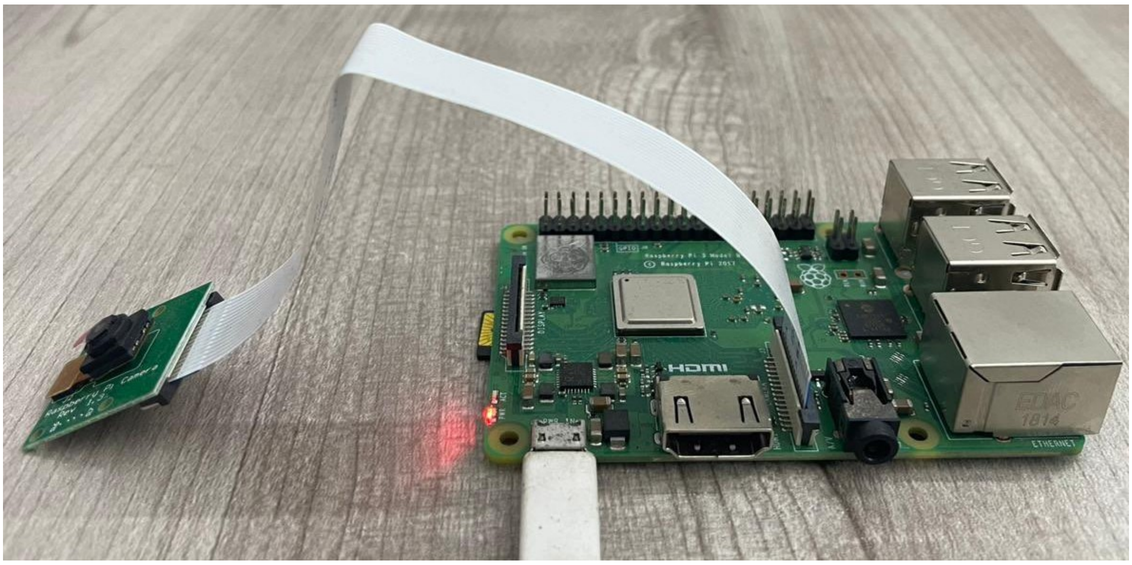


Figure 6.2: Raspberry Pi Model

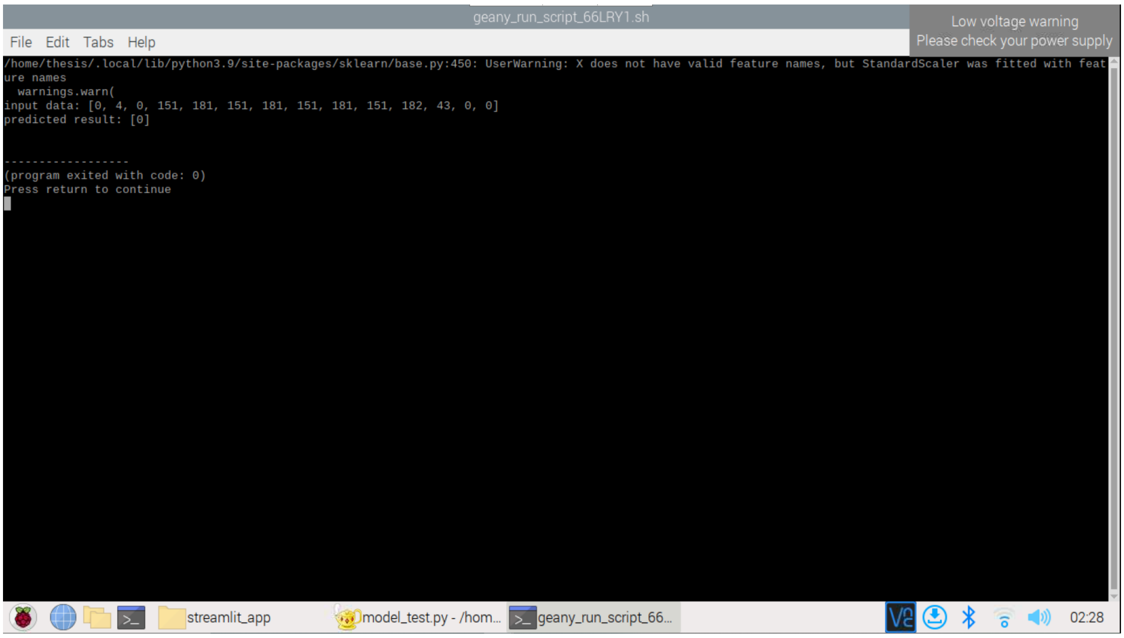


Figure 6.3: Raspberry Pi Terminal

Deep learning models often require significant computational resources, particularly in terms of GPU power. The Raspberry Pi, while a capable single-board computer, may not have the necessary hardware components to handle the demands of running a deep learning model. Additionally, cloud services such as Amazon Web Services or Google Cloud Platform offer powerful GPU instances that can be used to train and run deep learning models, making them a more viable option for this type of task.

Therefore, we have deployed a classification system for our image dataset using flask and TensorFlow Lite. Our model is able to correctly classify images as normal or abnormal and a user-friendly interface has been created to interact with the model. However, limitations include the lack of access to high computational GPU edge devices and cloud services. Future research can focus on utilizing high computational GPU edge devices and cloud services to improve the performance of the model. The figure 6.4 demonstrates the deployment of the image classification model.

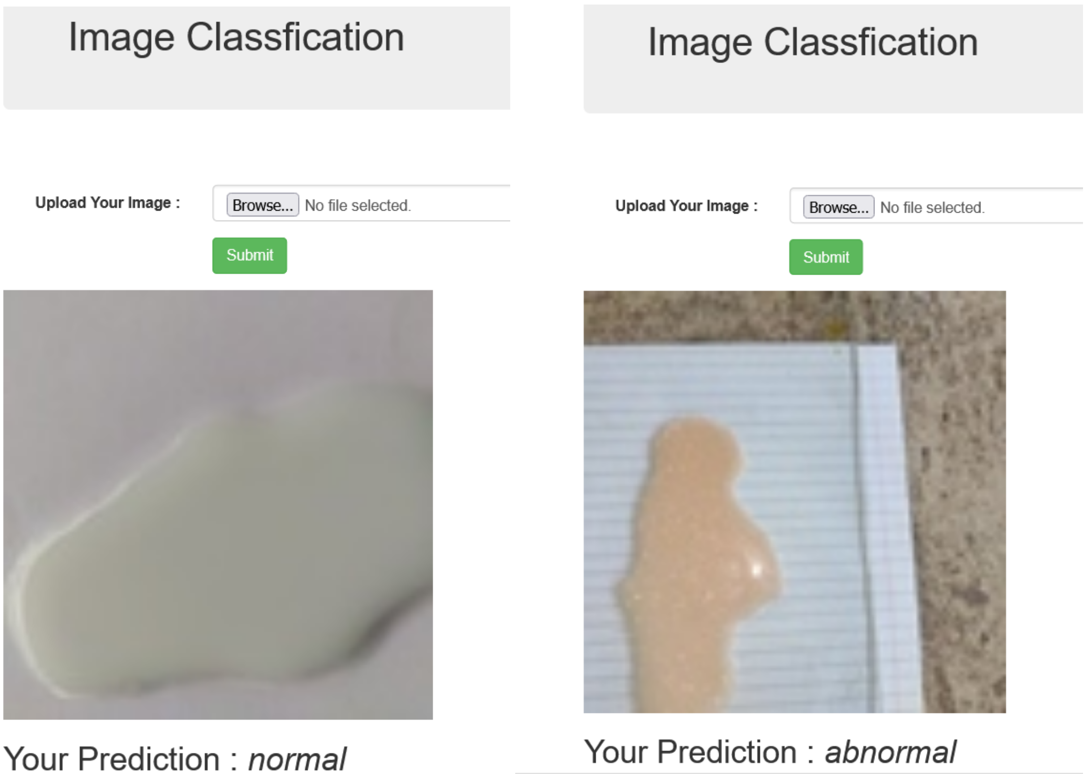


Figure 6.4: Image Classification

The following figure 6.5 demonstrates a proposed pipeline demonstrates a system that processes and analyzes images using deep learning techniques and makes the output available through a REST API for use on the Raspberry Pi device. After acquiring the image as input through the device, it is then preprocessed to ensure optimal performance. The image is passed through an AWS data pipeline utilizing cloud services for efficient and secure transfer of data. The image is transformed into a Docker image and a virtual environment is created using EC2 and other necessary resources.

Later the deep learning model (inceptionV3) is deployed on the virtual machine, utilizing high computational GPU capabilities for improved performance. The data is preprocessed with proper training and testing methods and fine-tuned using AWS SageMaker. The resulting model is stored in the model registry and made available through an AWS SageMaker endpoint. The final output is then converted into a REST API for easy integration with the Raspberry Pi device. AWS services such as the data pipeline, EC2, SageMaker, and the model registry are used to efficiently transfer and process large amounts of data, as well as deploy and serve deep learning models. The pipeline is highly efficient and effective in processing and analyzing images, and the final output being made available as a REST API ensures easy integration with other systems and devices.

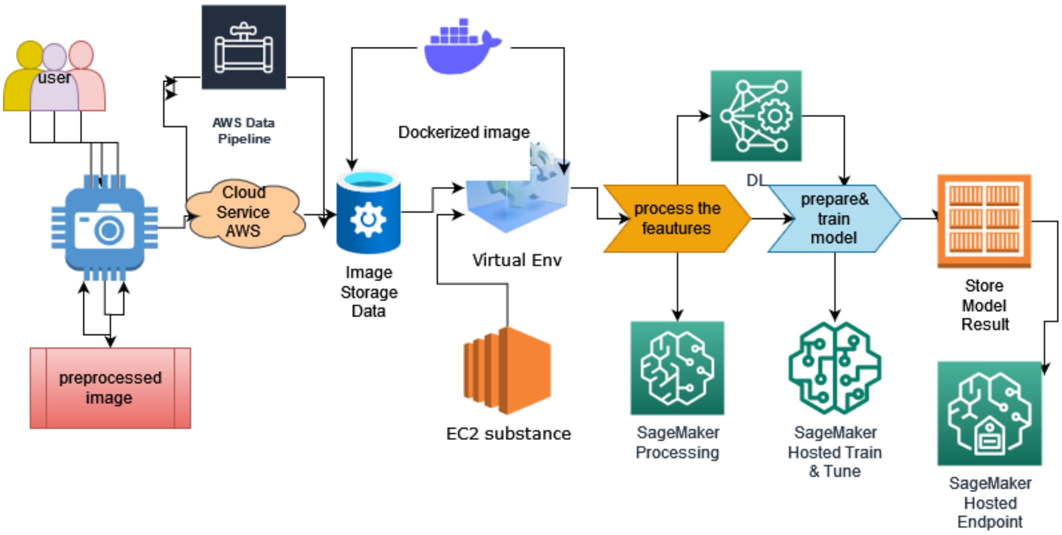


Figure 6.5: Cloud Infrastructure of the proposed future model

Chapter 7

Future work and Conclusion

7.1 Conclusion

In conclusion, the proposed system for detecting bovine mastitis in livestock using Deep Learning and Machine Learning techniques is a promising solution to address the economic impact of this disease in the dairy industry of Bangladesh and other developing countries. The system aims to provide an accurate and prompt diagnosis of mastitis which ultimately reduces costs and improves the efficiency of treatment. The system utilizes Deep Learning for image classification and Machine Learning for sensor data classification and is based on edge devices that allow for real-time data collection and analysis. This system has the potential to significantly reduce the economic impact of mastitis in the dairy industry of Bangladesh and other developing countries by providing a timely and accurate diagnosis, which in turn can help to improve treatment efficiency and protect the health and productivity of livestock animals.

7.2 Future Work

In the future, we aim to improve the accuracy and efficiency of our mastitis detection system by using real-world data and incorporating advanced technologies such as reinforcement learning and blockchain. By exploring the use of reinforcement learning to train the Machine Learning model more accurate predictions of detecting mastitis can be achieved. This will involve adjusting the model's parameters based on feedback from the cows' vital signs and other data and using an automation system to perform tasks on the dairy farm. Reinforcement learning can also be used to optimize the performance of the Deep Learning model by training it to make more accurate predictions based on feedback from the cows' images. By using blockchain technology to secure the data collected by the IoT devices a secure and transparent system for sharing data between different stakeholders can be created. This will involve storing the data on a decentralized ledger, creating digital identities for the cows, and creating a decentralized platform where farmers can share data with veterinarians and researchers. Blockchain technology can also be used to automate tracking and recording data. The integration of these advanced technologies will enable us to improve the overall accuracy and efficiency of our mastitis detection system, making it more useful for farmers, veterinarians, and researchers alike.

Bibliography

[1] I. S. Faruq, A. S. Ahsan, M. U. H. A. Rasyid, and Y. P. Pratama, "Health monitoring and early diseases detection on dairy cow based on internet of things and intelligent system," pp. 183–188, 2019. DOI: 10.1109/ELECSYM.2019.8901527.

[2] C. Vuppalapati *et al.*, "Artificial intelligence (ai) infused cow necklace - for diagnosis of bovine respiratory diseases," in *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, 2018, pp. 222–229. DOI: 10.1109/ICMLC.2018.8526937.

[3] A. Murphy, N. Creedon, A. O'Riordan, and I. O'Connell, "Data acquisition for ultra micro and bovine respiratory disease electrochemical immunosensors," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1–5. DOI: 10.1109/ISCAS45731.2020.9181288.

[4] A. Mohan, R. D. Raju, and P. Janarthanan, "Animals disease diagnosis expert system using convolutional neural networks," in *2019 International Conference on Intelligent Sustainable Systems (ICISS)*, 2019, pp. 441–446. DOI: 10.1109/ISS1.2019.8908108.

[5] M. Taneja, J. Byabazaire, A. Davy, and C. Olariu, "Fog assisted application support for animal behavior analysis and health monitoring in dairy farming," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, pp. 819–824. DOI: 10.1109/WF-IoT.2018.8355141.

[6] S. Ma *et al.*, "Development of non contact body temperature monitoring and prediction system for livestock cattle," *IEEE Sensors Journal*, vol. 21, no. 7, pp. 9367–9376, 2021. DOI: 10.1109/JSEN.2021.3056112.

[7] S. M. C. Porto, F. Valenti, G. Castagnolo, and G. Cascone, "A low power gps-based device to develop kde analyses for managing herd in extensive livestock systems," in *2021 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, 2021, pp. 243–247. DOI: 10.1109/MetroAgriFor52389.2021.9628711.

[8] F. Giametta, P. Catalano, G. La Fianza, and A. Simoni, "Bovine rfid tracing system with livestock safety remote sensing," in *2011 IEEE/SICE International Symposium on System Integration (SII)*, 2011, pp. 168–171. DOI: 10.1109/SII.2011.6147439.

[9] K. Stański, S. Lycett, T. Porphyre, *et al.*, "Using machine learning improves predictions of herd-level bovine tuberculosis breakdowns in great britain," *Sci Rep*, vol. 11, p. 2208, 2021. DOI: 10.1038/s41598-021-81716-4.

[10] S. M. Rizanov and P. I. Yakimov, "Cattle disease detecting iot thermographic system," pp. 1–6, 2021. DOI: 10.1109/ET52713.2021.9579577.

[11] A. Datta *et al.*, "In-body to out-of-body communication channel modeling for ruminant animals for smart animal agriculture," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, IEEE, 2021, pp. 7570–7573. DOI: 10.1109/EMBC46164.2021.9629743.

[12] A. A. Chaudhry, R. Mumtaz, S. M. Hassan Zaidi, M. A. Tahir, and S. H. Muzammil School, "Internet of things (iot) and machine learning (ml) enabled livestock monitoring," in *2020 IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*, 2020, pp. 151–155. DOI: 10.1109/HONET50430.2020.9322666.

[13] C. w. Maina, "Iot at the grassroots — exploring the use of sensors for livestock monitoring," in *2017 IST-Africa Week Conference (IST-Africa)*, 2017, pp. 1–8. DOI: 10.23919/ISTAFRICA.2017.8102356.

[14] N. Ma, L. Pan, S. Chen, and B. Liu, "Nb-iot estrus detection system of dairy cows based on lstm networks," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, 2020, pp. 1–5. DOI: 10.1109/PIMRC48278.2020.9217214.

[15] E. Bocaj, D. Uzunidis, P. Kasnesis, and C. Z. Patrikakis, "On the benefits of deep convolutional neural networks on animal activity recognition," in *2020 International Conference on Smart Systems and Technologies (SST)*, 2020, pp. 83–88. DOI: 10.1109/SST49455.2020.9263702.

[16] Y. Feng *et al.*, "Socialcattle: Iot-based mastitis detection and control through social cattle behavior sensing in smart farms," *IEEE Internet of Things Journal*, DOI: 10.1109/JIOT.2021.3122341.

[17] O. Unold *et al.*, "Iot-based cow health monitoring system," in *Computational Science - ICCS 2020. ICCS 2020. Lecture Notes in Computer Science*, vol. 12141, Springer, 2020. DOI: 10.1007/978-3-030-50426-7_26.

[18] E. Tanyildizl and G. Yildirim, "Performance comparison of classification algorithms for the diagnosis of mastitis disease in dairy animals," in *7th International Symposium on Digital Forensics and Security (ISDFS)*, 2019, pp. 1–4. DOI: 10.1109/ISDFS.2019.8757469.

[19] K. Ankitha and D. Manjaiah, "Comparison of KNN and SVM Algorithms to Detect Clinical Mastitis in Cows Using Internet of Animal Health Things," in *Intelligent Data Engineering and Analytics*, S. Satapathy, Y. Zhang, V. Bhateja, and R. Majhi, Eds., ser. Advances in Intelligent Systems and Computing, vol. 1177, Singapore: Springer, 2021, pp. 61–68. DOI: 10.1007/978-981-15-5679-1_6.

[20] T. Bobbo, S. Biffani, C. Taccioli, and et al., "Comparison of machine learning methods to predict udder health status based on somatic cell counts in dairy cows," *Scientific Reports*, vol. 11, no. 1, p. 13 642, 2021. DOI: 10.1038/s41598-021-93056-4.

[21] F. Vique, H. Marichal, and L. Steinfeld, “Inline mastitis detection system measuring the electrical conductivity of quarter milk,” in *2020 IEEE International Conference on Industrial Technology (ICIT)*, 2020, pp. 777–784. DOI: 10.1109/ICIT45562.2020.9067172.

[22] S. Kimura, J. Fukuda, and H. Suzuki, “Diagnostic device for cow mastitis based on the detection of superoxide secreted from neutrophils,” in *2011 16th International Solid-State Sensors, Actuators and Microsystems Conference*, 2011, pp. 775–778. DOI: 10.1109/TRANSDUCERS.2011.5969309.

[23] K. L. S. Tungol, J. M. B. Primero, and J. F. Villaverde, “Detection of bovine mastitis milk through somatic cell count using electrical conductivity property,” in *2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, 2020, pp. 1–6. DOI: 10.1109/HNICEM51456.2020.9400084.

[24] M. Bari, M. Alam, M. Uddin, and M. K. Rahman, “Prevalence and associated risk factors of bovine clinical mastitis in patiya upazila under chittagong district of bangladesh,” *International Journal of Natural Sciences*, vol. 4, no. 5, 2016. DOI: 10.3329/ijns.v4i1.28585.

[25] K. Liu, S. Yu, and S. Liu, “An improved inceptionv3 network for obscured ship classification in remote sensing images,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 4738–4747, 2020. DOI: 10.1109/JSTARS.2020.3017676.

[26] A. Datta, M. Haider, and S. Ghosh, “Economic analysis of dairy farming in bangladesh,” *Tropical Animal Health and Production*, vol. 51, 10.1007/s11250-018-1659–7, 2019.

[27] Z. Dongmei, W. Ke, G. Hongbo, W. Peng, W. Chao, and P. Shaofeng, “Classification and identification of citrus pests based on inceptionv3 convolutional neural network and migration learning,” in *2020 International Conference on Internet of Things and Intelligent Applications (ITIA)*, 2020, pp. 1–7. DOI: 10.1109/ITIA50152.2020.9312359.

[28] Z. Bao, Y. Wu, G. Pan, Y. Liu, J. Hu, and Y. Liu, “Vgg16 fire image fast detection based on multi-layer features fusion,” in *2022 IEEE 5th International Conference on Electronics Technology (ICET)*, 2022, pp. 1158–1162. DOI: 10.1109/ICET55676.2022.9824459.

[29] A. K. Abdar, S. M. Sadjadi, H. Soltanian-Zadeh, A. Bashirgonbadi, and M. Naghibi, “Automatic detection of coronavirus (covid-19) from chest ct images using vgg16-based deep-learning,” in *2020 27th National and 5th International Iranian Conference on Biomedical Engineering (ICBME)*, 2020, pp. 212–216. DOI: 10.1109/ICBME51989.2020.9319326.

[30] A. Panthakkan, S. M. Anzar, S. A. Mansoori, and H. A. Ahmad, “Accurate prediction of covid-19 (+) using ai deep vgg16 model,” in *2020 3rd International Conference on Signal Processing and Information Security (ICSPIS)*, 2020, pp. 1–4. DOI: 10.1109/ICSPIS51252.2020.9340145.

[31] Y. Zhao, K. Xie, Z. Zou, and J.-B. He, “Intelligent recognition of fatigue and sleepiness based on inceptionv3-lstm via multi-feature fusion,” *IEEE Access*, vol. 8, pp. 144 205–144 217, 2020. DOI: 10.1109/ACCESS.2020.3014508.

[32] W. N. Cheng and S. G. Han, “Bovine mastitis: Risk factors, therapeutic strategies, and alternative treatments - a review,” *Asian-Australas J Anim Sci*, vol. 33, no. 11, pp. 1699–1713, 2020. DOI: 10.5713/ajas.20.0156.

[33] R. T. Dingwell, K. E. Leslie, Y. H. Schukken, J. M. Sargeant, and L. L. Timms, “Evaluation of the california mastitis test to detect an intramammary infection with a major pathogen in early lactation dairy cows,” *Canadian Veterinary Journal*, vol. 44, no. 5, pp. 413–415, 2003.

[34] A. K, M. D H, and K. M, *Data for: Clinical mastitis in cows based on udder parameter using internet of things (iot)*, 2020. DOI: 10.17632/kbvcdw5b4m.1. [Online]. Available: https://data.mendeley.com/datasets/kbvcdw5b4m/1.