Application-Specific Transfer Learning over Edge Networks

by

Deepak Saggu

A thesis submitted to the
School of Graduate and Postdoctoral Studies in partial
fulfillment of the requirements for the degree of

Master of Applied Science in Electrical and Computer Engineering

Faculty of Engineering and Applied Science

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

November 2021

THESIS EXAMINATION INFORMATION

Submitted by: **Deepak Saggu**

Master of Applied Science in **Electrical and Computer Engineering**

Thesis Title: Application - Specific Transfer Learning over Edge Networks

An oral defense of this thesis took place on October 28, 2021 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee: **Dr. Khalid Elgazzar**

Research Supervisor: **Dr. Akramul Azim**

Committee Member: **Dr. Masoud Makerehch**

Thesis Examiner: **Dr. Omar Alam**

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

ABSTRACT

Transfer learning uses a profound labeled set of data from the source domain to deal with a similar problem for the target domain. Transfer learning provides accurate decision-making when insufficient data samples are available and when building a new prediction model takes more time and effort. This study explains comparative analysis of traditional machine learning techniques and transfer learning approaches over edge networks to enhance the performance and networking latency within discrete nodes. Edge networks are widely used to improve the efficiency and staging of any algorithm as the embedded systems focus on implementing some particular events based on the microprocessors and, at the same time, working on the least resources that result in having less power consumption. Moreover, we generated a hybrid-based transfer learning model to avoid negative transfer. This thesis uses two case studies: mushroom sales prediction and heart attack detection system.

Keywords: Machine Learning; Transfer Learning; Embedded Systems; Edge Networks; Classification and Regression

AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the Ontario Tech University to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize Ontario Tech University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.

Deepak Saggu

STATEMENT OF CONTRIBUTIONS

The contributions of my thesis are listed as follows:
a) We implement the comparative analysis of various traditional machine learning algorithms for the classification problem, showing the accuracy scores, and compare the results with homogeneous transfer learning approaches. Also, we introduce the way to avoid negative transfer using a hybrid-based transfer learning approach.
b) We introduce a faster way of implementing transfer learning techniques, i.e., transfer learning on the edge networks, and we propose the system architecture that shows the networking of edge networks.
c) We experimentally prove that the transfer learning over the edge networks is working more efficiently than the traditional machine learning technique compiling over edge devices.

One of my contribution that has already been published is mentioned below:

- Deepak Saggu, and Akramul Azim. "Transfer Learning on the Edge Networks." 2021 IEEE International Systems Conference (SysCon). IEEE, 2021.

ACKNOWLEDGEMENTS

I want to thank Ontario Tech University for allowing me to explore my abilities and completing my thesis that will be helpful in the better of future technologies.

The completion of this thesis could not have been possible without the help and assistance of Dr. Akramul Azim, Associate Professor, Ontario Tech University. He has always provided thoughtful solutions, guidance, and valuable comments throughout the learning process of my master's thesis. Dr. Azim was always available for all inquiries, even in the shortest period. He always motivated me to work towards my interests and directed me in the right direction. I appreciate that you have faith in me, and thank you for all your time and efforts.

Moreover, I would like to acknowledge the help of all my RTEMSOFT research group members for supporting and helping me though-out my courses and research work. I want to thank you all for providing honest feedback on all the research work that I have presented during the completion process of my thesis. I would also like to thank all the respected faculty members of Ontario Tech University for providing extra support and assistance.

Finally, I want to express my gratitude towards my family, especially my brother Sandeep Singh Saggu, my mother Paramjit Kaur, my father Jagraj Singh, and my sister-in-law Harmeet Kaur for their continuous support and encouragement during all the years of my education. This achievement could not have been possible without my family, and I am very thankful to everyone.

Table of Contents

List of Figures

Abbreviations

ML	Machine Learning
DL	Deep Learning
TL	Transfer Learning
SVM	Support Vector Machine
KNN	k-Nearest Neighbours
LSTM	Long Short-Term Memory
PCA	Principal Component Analysis
BPNN	Back-Propagation Neural Networks
DNN	Deep Neural Networks
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
Iot	Internet of Things
MEC	Multi-access Edge Computing
NB	Naive Bayes Classifier
MA	Moving Average
TML	Traditional Machine Learning
CNN	Convolutional Neural Network
UFL	Unsupervised Feature Learning
SIFT	Scale Invariant Feature Transformation
RNN	Recurrent Neural Network

Chapter 1

Introduction

In machine learning, many researchers came up with numerous innovations to improve existing machine learning technologies and remarkably implemented them in real-life applications. Machine Learning is the most accurate method for analyzing and leveraging the dependencies and patterns in complex datasets [3]. Supervised machine learning models are trained on labels that are dependent on various features and attributes [4]. Labels can either be binary/ classification ideals or predictive/ regression ideals. Whereas, for unsupervised problems, the training of the machine learning model can be complex due to insufficient labeled data samples. However, traditional machine learning still has some limitations in certain areas of research. The basic schema of using a machine learning model has large labeled datasets similar to the testing data samples.

Nevertheless, it is challenging to obtain enough training data samples for safety-critical domains and some other real-life scenarios, and it can be computationally expensive. The feature space of the dataset can be different for the training and testing set as training samples may have a particular group of attributes, and the testing set may have additional features. The training and testing samples can be different based on probability distribution. For these types of differences among data, it is necessary to train a model on new training labeled data and validate the testing set. Traditional machine learning methods often predict visual data using mathematical and statistical models trained on previous data sets. The stated action plan performs only when training and testing data samples fall under equal distribution in the same feature area. The system should have labeled data points for training a model, but not all models have sufficient labeled data sets. Transfer learning is the best way to tackle this issue; the purpose is to transfer the knowledge using adequate details from the previous data of the source domain. The components or objectives can be different but associated, and there has to be some relationship; otherwise, a transfer is impossible. Supervised data samples are essential for source work, but job classification should be the same for both problems. Although one can solve this issue using a semi-supervised learning approach, it requires only a few labeled data points and many unlabeled data samples. In some scenarios, getting enough unlabeled datasets is also challenging. These limitations are considered for improvement in the traditional machine learning approach. It requires a model with high performance that can work regardless of the differences among the two datasets, which is possible using transfer learning techniques.

The machine learning model tends to produce good prediction outcomes based on the type of the desired outcome. We will use LSTM (long short term memory) as the pre-trained system model for performing transfer learning and predicting mushroom sales. Moreover, computing the RMSE and MSE scores of the model will depict the efficiency of the implemented model. LSTM technique works well to solve the prediction problem when the outcome depends on the input variables. We will be using the same model for implementing traditional machine learning as well.

Moreover, for another case study, heart attack detection, we will be using the logistic regression model for the binary classification problem of this medical domain. We implemented logistic regression directly for performing the traditional machine learning and computed the accuracy of the model. Furthermore, breast cancer prediction is the source domain trained on the Logistic regression. It will act as the pre-trained model for heart attack detection for implementing the feature-based transfer learning approach.

Transfer learning, and advancement in a field using the knowledge gained from any other corresponding field, enhances progress and improves the model's performance. The term transfer learning is a part of machine learning that includes the re-usage of any algorithm or model as a starting point for the new problem. As humans learn from experiences and use that learning in different ways, transfer learning is the same concept working on generalizing the experiences and implementing those experiences in the form of learning to solve the new problems. Every model or idea has its requirements and constraints. Transferring the knowledge amongst two situations is only possible if there is any relation between those two learning situations. For instance, in the general world, a human who knows how to ride a two-wheeler motor vehicle can learn faster that how to drive a car, as both situations have some standard features like rules, judgments, and adjustments for the road. Figure 1.1 depicts the essential elements of transfer learning. Source data refers to the source domain that is associated with the target domain. Both of the fields are related in terms of some generic similarities. Source domain helps to improve the learning system of the target domain by transferring the knowledge of the system model.

With the help of transfer learning concept, the data samples that are not sufficient for training only require pruning and fine-tuning of the pre-trained model without starting from the beginning.

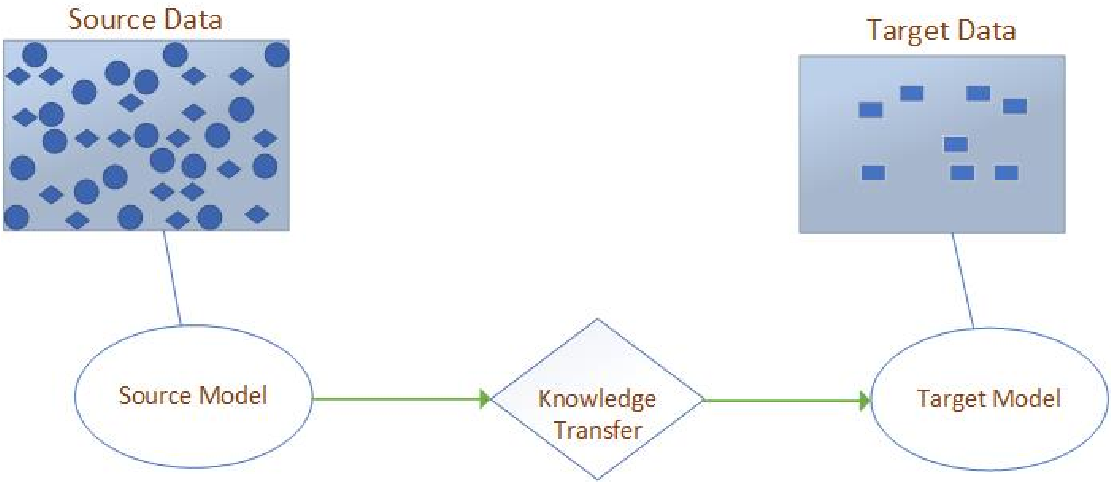

FIGURE 1.1: Transfer learning

Conforming to the differences in the source and target domain, transfer learning consists of two subcategories: Homogeneous transfer learning and Heterogeneous transfer learning [5]. Homogeneous transfer learning methods are generated to tackle the domains that belong to the identical feature space. Few studies conclude that environments contradict only in marginal distributions for homogeneous approach[6]. Hence, they adopt the domains by rectifying the sample selection bias or covariate shift. These domains may have similar attributes/ features space as represented ($X_s = X_t$), similar labels space ($Y_s = Y_t$). It focuses on leveraging the breach of distribution amongst two domains[6]. Homogeneous transfer learning has various approaches to solve the problems listed as Instance-based, parameter-based, symmetric, or asymmetric feature-based approaches and hybrid-based approaches (combination of two approaches) [6]. Nowadays, almost all projects are associated with homogeneous domain tasks. The purpose of using the above-listed approaches is to decrease the performance loss because of the dispensation among data samples. Heterogeneous transfer learning has different feature space, label space, or probability distributions amongst two domains. It can be represented as ($X_s \neq X_t$), or ($Y_s \neq Y_t$); in this case, it is challenging to leverage and analyze the cross-domain distribution gap and relationship between the two environments. The models working under a heterogeneous data approach might result in a negative transfer.

It is not always true that the similarities between the two fields will only give good results. It can also be a negative transfer. There can be a point where a negative transfer can occur, [7] describes the methodology that can delay the process of negative transfer at a specific point during the initiation state. This method can also stop the transfer learning approach when it does not affect the performance. Negative transfer is an issue when knowledge transferred from the source field reduces the accuracy level of the target task. Negative transfer occurs when the target and the source domains have more differences rather than similarities. It is possible in both homogeneous and heterogeneous transfer learning methods. For instance, if one has to develop a primary classifier using the finite amount of supervised target data samples, then a primary classifier would work better than the transfer learning model that has been pre-trained on the source training data points. To reduce negative transfer, there has to be some similarities between the two situations, the more related they are, the less cross-domain distribution gap will be. As discussed in this 4 section, a negative transfer occurs while solving the regression problem on mushroom demand prediction using the feature-based approach, we solved it by using a hybrid-based system that is the integration of feature and parameter-based techniques, and this method gave us better results than feature-based approach. In this scenario, we are using the source domain as avocado price prediction.

We propose an accelerated way of applying different transfer learning approaches for prediction and classification problems on edge networks. Edge networks are widely used everywhere to maximize the efficiency of any learning model. Embedded systems focus on one event or limited stated functions. They are extensively used far and wide, such as portable audio players, cruise controls, cell phones, house appliances, etc. Embedded systems are also used in different machine learning programs to implement many differentiating and predictable events successfully. Many industries run multiple portfolios of work over several computer systems using edge devices, resulting in operational delays and more computation power. Installing more local devices is not an economically good solution. Users can import an embedded device networked with the existing independent system and immediately run specific tasks to deal with this issue.

This study will perform traditional machine learning approaches, homogeneous transfer learning approaches, and edge networks on two different case studies. Case study one is the food demand and sale prediction, and second is the safety-critical medical domain.

1.1 Problem Statement

Nowadays, almost every industry area requires some statistics performed on their systems to deal with the prediction and classifying problems. Traditional machine learning techniques work great to address these kinds of tasks. Any ML technique prefers a decent volume of previous data/information to train and validate the system model. Some domains do not have enough data samples available for the training of the model. In these cases, the results can be fluctuating and not be accurate.

Moreover, sometimes if one has enough sample points for training, there can be sparsity in the data, resulting in multiple null values, affecting the system's performance. For addressing these obstacles, we use the concept of transfer learning that solves the sparse data and performance lacking issues and gives a better prediction in less time. The training samples only have to perform the fine-tuning on the pre-trained network model and do not need to get trained from the beginning.

Furthermore, there can be some industries or domains where the model might take some time to compile and generate the outcomes due to more extensive datasets or storage problems. Nowadays, various projects work simultaneously, such as running data testing, analyzing, forecasting using multiple machine learning techniques. However, when the user requires to compile another program over the same device, that might take more computational time to do that project, and there may be some operational delays. For agricultural-based projects, when there is no internet connectivity and have less budget, then we can introduce edge networking. As engineers can install a low power and resource constraint device(embedded system) linked with their existing system, this will result in giving efficient outcomes using fewer resources and being more cost-effective.

1.2 Novelty of thesis

The novelty of this work focuses on implementing the hybrid-based transfer learning approach on regression problems to avoid negative transfer in the best efficient way in terms of performance, accuracy, and errors. Moreover, we are accelerating the performance of our proposed method by applying it over the edge networks that result in compiling up the output even faster. We use two case studies to validate the methodology; mushroom demand prediction and heart attack detection system.

The mushroom demand prediction domain addresses the regression problem. We will use the LSTM (Long-short term memory) model already trained on the previous related domain; avocado price prediction. The data for this scenario is the actual data. It has been collected from the mushroom farm in Ontario that depicts their sales for six months and has various features like the type of mushrooms, customer ID, customer name, date, day, and quantity.

The heart attack detection system addresses the binary distribution problem. We will use the logistic regression technique to classify whether there is a possibility of having a heart attack or not. The source domain for this environment is breast cancer prediction with binary labels, and it is a large dataset working great using logistic regression. This scenario is a pure homogeneous transfer learning problem as its feature space and label space are very similar.

Both domains are crucial in today's world of technology; the heart attack detection system comes under the safety-critical software systems. It demands a lot of historical data and requires further validation for the results, and collecting the basic historical information for training has various consent requirements and legalizations. For the food industry, as it is developing forward in technology to forecast the demand and sales to produce the approximately accurate production, perishable items like mushrooms need technical attention that might save some time and finances for the industry owners.

Moreover, for even accelerating the performance of both the system models, we will be implementing them over the edge networks as it will cut off some computational latencies and work for only particular system models. We will be connecting an edge device to the embedded system to form an edge network; the model should be implemented on the embedded system and store the data samples for training and testing in the edge device.

In that way, even if any corporation has only one edge device, they need to attach an embedded system and compile the system model on it. That will be inexpensive compared to setting up the whole new edge device.

1.3 Contributions

In summary, the main contributions of this thesis are:

1. We perform the comparative analysis of the traditional machine learning methods on regression and classification problems with the homogeneous transfer learning approaches. Moreover, we acknowledge the concept of negative transfer using a feature-based transfer learning approach and avoid the negative transfer using hybrid-based transfer learning on the same domain.

2. We propose an accelerating way of using transfer learning, i.e., transfer learning on the edge networks [8]. We offer a figure that depicts the system-model architecture for an edge network and shows how the connection works.

3. Furthermore, we experimentally prove that applying transfer learning approaches on the edge networks works better than compiling it over the edge device. The homogeneous transfer learning approaches work better than the fundamental machine learning methods for regression and classification problems.

1.4 Organization of the thesis

We categorize our thesis into five components. Chapter 2 consider the literature review that discusses various concepts and facts on fundamental machine learning, categorical transfer learning, embedded systems, and edge networks. This section discusses some existing methodologies working on regression and classification problems to apply transfer learning.

In Chapter 3, we present the proposed approaches of the thesis that has different sections for better understanding. We define the system model and system architecture. Hence, we perform the comparative survey of machine learning techniques with the transfer learning approaches. Also, we propose the methodology to avoid negative transfer happening on a feature-based system by applying the hybrid-based TL approach. Moreover, we present the method of implementing transfer learning models over edge networks.

In Chapter 4, we present the experimental analysis of the thesis that has different sections to better understanding the methodology implemented step by step. We use the two case scenarios for the experiments, i.e., Mushroom demand prediction and Heart attack detection system. WE provide the results figures with the accuracy value and the RMSE and MSE scores to understand the clear comparison. Moreover, we mention all the delays and latency happening over edge devices and edge networks.
We prove that hybrid-based transfer learning helps in avoiding the negative transfer. Also, the transfer learning approach works better than the fundamental machine learning techniques, and executing proposed transfer learning techniques on the Edge networks accelerates the performance of the system model.

At last, Chapter 5 concludes the thesis and points out possible future research directions in the context of transfer learning and avoidance of computational delays while using transfer learning approaches.

Chapter 2

Literature review

2.1 Traditional Machine learning

Future prediction of sales in the food companies like groceries, superstores, food chains is very concerned. The author applied various machine learning approaches to predict food sales [9]. This study enlightens the input variables necessary for the anticipation of sales and the type of output variables. They performed the survey on applying different machine learning models on demand and sales prediction problems.

The accuracy of sales prediction contributes a significant role in business. Data mining methods are very efficient ways of exposing the hidden features from a large dataset to increase the performance of the decision. The explained research and reasoning of prediction techniques to enhance the demand anticipation are carried out in this study[10]. Existing prediction systems are not easy to work with extensive data and do not provide efficient results. This study has overcome the above issue by using other data mining methods. This study focused on the analysis of the sales dataset and its prediction. According to the performance results, an appropriate predictive technique is advised for the prediction of sales patterns. The outcomes are explained based on reliability and efficiency. The research found the best fitting algorithm is Gradient Boost Algorithm that depicts the most certainty in forecasting the future sales [11].
Smart farming involves the integration of predicting algorithms into existing agriculture techniques to increase production quality. It is better leveraged with data analysis techniques, ML and IoT. A central concept of smart farming is to manage productivity using advanced technologies that allow the production in large quantities according to the demand. For instance, machine learning influences agricultural demand prediction

to make accurate decisions on crop production supply. Smart farming helps analyze and predict the future direction, including various factors like weather and market economic conditions based on the existing historical data of sold crops.

Sales expectations are a complex issue because of the many factors that affect the product. We present a predictive sales forecast method that recognizes content and selects a basic prediction based on historical sales structure. The experimental section shows that product subsets are present where non-knowledgeable routes can pass. We also provide dependence between product segmentation details and the accuracy of sales forecasts [12]. An experiment conducted at a food company shows that predictable moving averages can be made more innovative in a fair separation, which seems to be a complex problem.

Data-based algorithms like traditional ML techniques and seasonal forecasting are used to predict sales in food manufacturing fields. However, there are not enough training data samples available for training the accurate systems for freshly introduced products[13]. Therefore, human interference machines are developed to enhance the efficiency of a prediction model. Human experts use their prior knowledge on domains having similar products for forecasting future sales. Hence, applying Transfer Learning introduced an analytical way to transfer the learning between existing and new fields of study. They develop a network-based transfer learning model for the deep convolutional network to check the system's performance. They conducted a case study on a fresh product based on a food manufacturing company from Australia for testing. The outcomes of the proposed approach show good results. [12].

Traditional machine learning techniques work well in real-life applications due to memory and technical enhancements. ML techniques are known for breaking down complex datasets and extracting the knowledge for dealing with science-related fields. Every algorithm has system requirements for the data samples and has some specific constraints that can be a limitation and not considered a good model. This study [14] describes the methodology to choose the correct method for the specific problem and data samples, and that makes the new machine learning algorithms work more efficiently.

For the labeled datasets, present input labels and respective output label space is provided for consideration. This information helps the machine to recognize the production for different inputs. The main supervised ML techniques are; KNN (K-nearest neighbor), SVM (Support Vector Machine), CNN, and Naive Bayes. The critical algorithms for unsupervised ML are; Principal Component Analysis (PCA), K-means clustering. Enhanced reading does not create inputs with consistent results [15]. This algorithm integrates with the learning environment. If its performance is high, the reward is given. Q-learning is an instance of reinforcement learning.

Every prediction model will give different prediction outcomes when used for one specific dataset. The MA (moving average) technique is well known for simple prediction problems. At the same time, LR (logistic regression) is used for classification problems. In contrast, the Back-Propagation Neural Network (BPNN) algorithm is suitable for long-term data samples.
In study [16], the author ran a comparative analysis of the performance based on three prediction ML models for anticipating sales in convenience stores using a POS database. An ML-based technique is the most recent and effective way to predict future data samples based on large sets of previous data. It gives them access to the user to analyze data by measuring a 'features' data group that primarily affects the outcome. Using these factors, the user can place the rules that follow the machine-learning approach and finally produce the result. The machine will compute the parameters and weights of all the features and determine the metrics that give the minimum error in the results. Then, several different techniques can be generated and installed to find those parameters.
This study compares traditional machine learning techniques with homogeneous transfer learning approaches over regression and classification problems. Machine learning offers numerous regression algorithms like the random forest, linear Regression, SVM (support vector machine), decision tree regression, etc. One of the regression algorithms is LSTM(long short-term memory) that works to classify and process the time series data and make predictions. LSTM is a neural network-based model. We use this model for one of the case studies that are mushroom demand prediction, as this domain has time-series data. Moreover, the second case study is a classification problem that is a Heart attack detection system, and this domain has categorical labels. However, machine learning provides various models to implement definite predictions like logistic regression, naive Bayes, K-nearest neighbors. We will be using Logistic Regression as this model gives discreet output labels and fits the sigmoid curve's line values. Also, LR (Logistic Regression) can handle large datasets, and this model is vulnerable to overfitting.

2.1.1 LSTM

A set of instances is measured consequently via time. Those observations are either non-stop through time at a discrete group of equal periods. Continuous-time series, including the count of mind activity, are generally analyzed to sample the collection at identical time durations to provide a discrete instance of time intervals. Since the archive will be the insight of discrete units of information, the relentless time series will presently don't be moreover dissected in this report. The observations at one-of-a-kind time series inside the array frequently relate to particular approaches. While recessive statements are dependent, and future values can also be anticipated from previous readings. A Time-series study is the study's vicinity in which the correlations and expectations are studied. The correlation may be the series of the captured facts, the model's linearity, recurring patterns, etc. Time-series evaluation affords strategies to analyze the statistics[17]. Time-collection forecasting entails using the time-series observations, collectively with the evaluation, to establish a version describing the dependencies.
RNNs, recurrent neural networks, chain together more than one layer of networks, wherein statistics from preceding time-steps, including the output, are carried out to destiny time-steps. According to the values and variables processed from every layer, the outcome from preceding layers is taken into thought, giving the community a form of reminiscence[18]. One baseline version becomes implemented the usage of a naive forecasting method. The model ganders at each worth in the time assortment and estimates the indistinguishable expense for the accompanying time-step.
The core concept of deriving equations of LSTM is based on backpropagation, cost function, and loss.

2.1.1.1 Notations

- W_f, b_f: Weight and Bias for Forget gate
- W_i, b_i: Weight and Bias for Input
- W_c, b_c: Weight and Bias for Cell state
- W_o, b_o: Weight and Bias for Output
- W_v, b_v: Related to the softmax layer
- f_t, i_t, o_t : Activation function's output

- a_f, a_i, a_c, a_o: Activation Function's input
- j : cost function

2.1.1.2 Equations

The following equation 2.1 focuses on the data that should be discarded or kept. Data from the previous hidden state and data from the present input is transmitted by sigmoid function. Numbers range from 0 and 1. Closer to 0 means forgetfulness, and closer to 1 means last.

$$\{a_f = (W_f.Z_f) + b_f\}where, \{f_t = sigmoid(a_f)\} \tag{2.1}$$

These 2.2 and 2.3 gates are also called save vector in terms of LSTM. Mentioned equations evaluates what information one should include in the cell/long-term memory. The most fundamental components are the functions of activating individual gates. The input gate consists of sigmoid function and has a distance of [0,1].

$$\{a_i = (W_i.Z_t) + b_i\}where, \{i_t = sigmoid(a_i)\} \tag{2.2}$$

$$\{a_c = (W_c.Z_t) + b_c\}where, \{\widetilde{(c_t)} = tanh(a_c)\} \tag{2.3}$$

The output gate 2.4 determines the value of the next hidden state. This state contains details about the previous installation. The Logistic Regression measures the relationship between diversity depending on diversity and one or more variables independently in terms of probability. Using a structured function, the distribution of the collection cite khairunnahar2019classification. Firstly, the present state and the primary hidden state variables are transported to the following sigmoid function. After that, a new cell structure formed from the cell is passed through the tanh function.

$$\{a_o = (W_o.Z_t) + b_o\}where, \{o_t = sigmaoid(a_o)\} \tag{2.4}$$

Cell state is also known as Long-term memory. The iterations represent the cell's recursive behavior, allowing data from prior instances to memorize within the LSTM cell. Cell state 2.5 is manipulated using the forget gate installed below the cell state and adjusted by the input modulation gate. Hidden state 2.6 is the outcome of the cell state.

$$\{C_t = f_t \bigotimes C_(t-1) \bigoplus i_t \bigotimes \widetilde(c_t)\} \tag{2.5}$$

$$\{h_t = o_t \bigotimes tanh(c_t)\} \tag{2.6}$$

2.1.2 Logistic Regression

The binary logistic regression model predicts classifying action depending on more predictor variables (e.g., features). It includes a supplementary probabilistic nature of classification, and this consists of the hypothesis, the cost function, and the sigmoid function. Every machine learning classifier is consists of various techniques like optimization, gradient descent algorithm, advanced optimization technique.
Along with that, ML techniques offer better performance with scaled features. The standard operating procedure refers to the re-balancing of objects by a specific measurement of short minutes; we must apply the min-max magnification process to every feature set.

2.1.2.1 Logistic Regression equations

$$X_{changed} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2.7}$$

The suspension centers around the even dissemination of the element sections to the mean zero and one deviation to make the standard typical dispersion of zero methods even with the variety of one. Learning algorithms can read instruments easily with the type of data used.

$$z = \frac{x - \mu}{\sigma} \tag{2.8}$$

Forwarding: Any CNN has a wide range of discussion, integration, and fully integrated layers. The composite layer contains a number of characters used to explore feature maps following a convolutional layer. The feature text of the textit j -th of textit l -th layer solution is calculated by:

$$a_j^l = \sum_{i=1}^{N^{l-1}} K_{ij}^l * x_i^{l-1} + b_j^l, x_j^l = f(a_j^l) \tag{2.9}$$

2.2 Transfer Learning & Edge Networks

Transfer learning (TL) [19] develops a better model for the target domain using limited and unlabeled training data points by extracting and analyzing the learning from a different yet related task domain and predict the labels for the unlabeled data. Due to insufficient label times, model training in this scoring challenge can lead to reduced performance compared with the system trained over inadequate separated figures. Although, by improving the training with additional labeled statistics from the corresponding distribution domain, the power of the translation variance of goal times can be improved. The project becomes a way to separate the beneficial understanding of the source domain from the natural sound of the domain due to the various distributions and placement of the goal. Transfer learning has two main classes in terms of the function spaces: homogeneous and heterogeneous transfer getting to know.

TL, or a negative transfer, indicates information transfer to a targeted activity from a source. Depending on the nature of convolutional neural networks, TL can be done by re-inserting layers of features learned from one CNN (derived from the source function) to implement another (target function). In [20], authors are investigating whether it is possible to be CNN's automatic backup source before using it for targeted operation. In particular, we present the framework of the conceptual framework to understand the direction of resource direction and use this as a basis for finding a way to measure CNN resources in a practical, non-judgmental way automatically.

Fine-tuning of the CNN model for the source domain might work better than training from the beginning. If you use fine-tuning, the bare supposition is that the previously trained system produces common attributes, at least slightly related to the resolution of the targeted task. Still, it can be challenging to subtract a restricted range of data samples obtained from the second domain. However, there is no way to properly prepare to retain the features you have learned in the source work without starting with a pre-trained model and pre-setting. In this study, we observe various practical schemes that enhance the resemblance of the final solution to the original system model [21].

The discovery of automatic grief separation and segregation has always been one of the top studies for research in the transportation industry. In [22], the author has hired a Deep Convolutional Neural Network (DCNN) trained in ImageNet's big data, that consist of a large number of image data samples. Moreover, he explores studying impulsively recognizing cracks in Hot-Mix Asphalt & Portland Cement Concrete stone sculptures that include inconsistent variability and disability. In addition to the familiar sources of false positives encountered in automated road-based observations, a more complex system was presented in this study to train differentiation in the combined HMA and PCC images with different facial features. One-layer network partition (with 'adam' user) trained in ImageNet's VGG-16-trained VNG-16 features has improved performance. This study [23] shows the transfer learning in edge computing for the proactive caching stuff in mobile devices. The goal of information acquisition is to improve the overall performance of the targeted business through a future/resource area. The problem will be that sometimes transferring information from a source domain can negatively affect the goal model, also known as a bad transfer. If a person creates a widespread fragmentation, the more detailed use of restricted, segregated software will have a higher performance than the transfer version using different objective and resource records. This is especially true when the source domain has very little in common with the target. With the transfer of success information, miles are thought to be the source and purpose of the relationship in several ways. Too little to do with it, the sound of going to a place has skills. As a result, more information is acquired and extracted that increases performance [24].

In [25], solutions to computer edge challenges and time-saving challenges in terms of power and latency are introduced. Temporary savings on men and comparisons between different methods of temporary storage in MENs have been presented. An example of a study conducted on the development of archives for wireless networking and learning technologies (ILTs) in a particular domain in its design has been presented. [6] confirmed this by displaying how shifting between various domains reasons performance loss. A few homogeneous transfers getting to know solutions employ safeguards towards terrible transfer as reviewed in. however, maximum current heterogeneous switch studying techniques no longer cope with this difficulty as to be later discussed. The Author [26] performs the comparative survey on the mobile edge networks using DL, ML over transfer learning.

- Inductive Transfer learning: Source and target domains spaces are similar, but the source and target ties are discrete from each other. Algorithms strive to use incoming discrimination of the source area to help enhance the targeted activity. Whether the service provider includes separate statistics or not, this will be equally

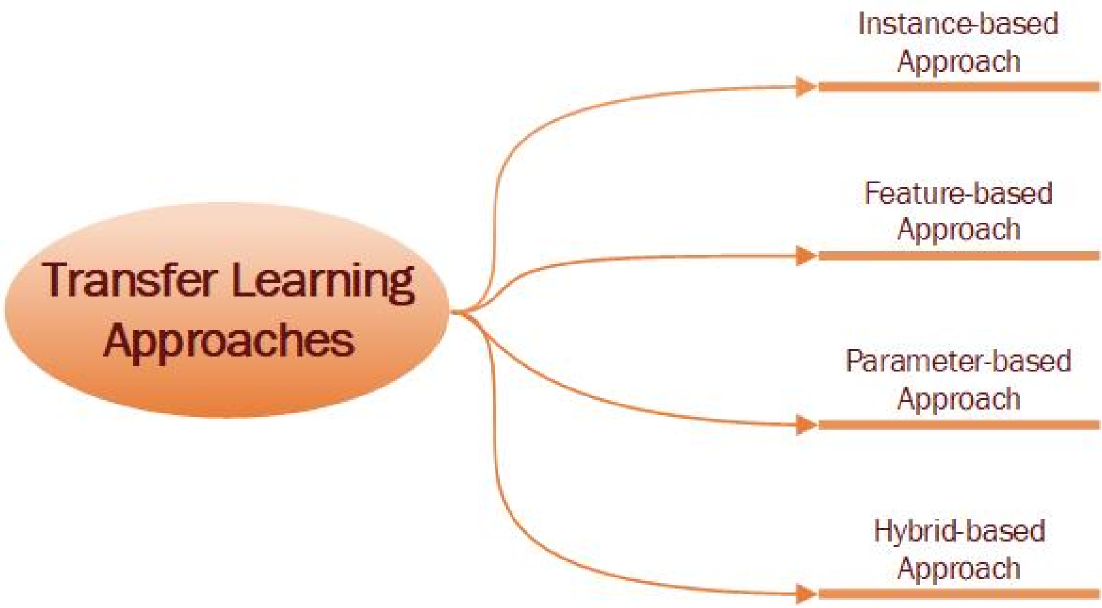

Figure 2.1: Categorization of Transfer Learning

divided into two sub-categories, such as multitask learning and self-teaching, respectively.

- Unsupervised Transfer Learning: This putting recites inductive TL, focusing on unsupervised obligations within the goal domain. The supply and goal domains are comparable, and however, the responsibilities are exclusive. In this scenario, classified information is unavailable in both of the domain names.

- Transductive Transfer Learning: This putting is like an inductive switch itself, focusing on unsupervised duties in the goal area. The supply and intention domain names are similar; however, the responsibilities are exclusive. In this scenario, classified information is unavailable in each of the domain names.

- Instance-based transfer: Commonly, using knowledge from the supply domain to the target mission is commonly a perfect situation. In most instances, the source area records can not be reused directly. Instead, specific models from the supply domain may be reused together with goal statistics to improve consequences [27]. In the case of inductive transfer, modifications together with AdaBoost by using Dai and their co-authors assist in using schooling times from the source domain for enhancements inside the target undertaking.

- Feature-representation based transfer: This method targets to decrease domain divergence and decrease blunders costs via figuring out excellent feature representations that can be utilized from the supply to target domain names. Depending upon the availability of categorized records, supervised or unsupervised methods may be carried out for function-illustration-based transfers.

- Parameter based transfer: This method focuses on the belief that the fashions for associated duties share a few parameters or previous distribution of hyperparameters. not like multitask learning, in which both the supply and target duties are discovered concurrently, for transfer learning, we might also practice additional weightage to the loss of the goal area to improve general overall performance.

Alongside,[28] study enlightens the concept of plant species using NN flexible mutations for transfer learning. The author used the pre-trained DNN models of AlexNet, GoogleNet, and VGGNet with outstanding network variables using data additions. They have integrated the parameters accompanied by excellent prediction settings to adjust the performance and functionality of the system. The results in the exemplary configuration of GoogleNet and VGGNet provide higher production than the ideal configuration of AlexNet. The paper additionally examined the most critical elements influencing efficiency: the number of iterations and the addition of data samples. One of the lessons [29] has worked on expanding systems, as it is a key element in generating any model, as well as making changes in goals. They introduced a well-working solution for reusing the experience to create a prediction system using in-depth learning methods. The creator presented an algorithmic methodology, known as the transfer learning-based dynamic multi-reason transformative calculation (EA), which coordinates transfer learning and human development calculations dependent on critical thinking multi-target improvement issues [30]. They exploit transfer strategies to improve the first practical model by reusing the experience to strengthen the evolutionary process. They validate their vision using an uncontrolled filtering algorithm, multi-purpose particles, and standard model-based measurements based on multiple distribution algorithm objectives.

Deep learning is a great way to extract relevant facts from uncooked sensory statistics from IoT gadgets used in complimented environments. Because of its versatile structure, deep knowledge is equally appropriate for a computer threshold. So, in this article, First we introduce an in-depth study of IoT in the computer environment. As pre-existing branches have restricted processing capacity, we have also developed a novel planning to improve the performance of in-depth computerized IoT learning applications [31].

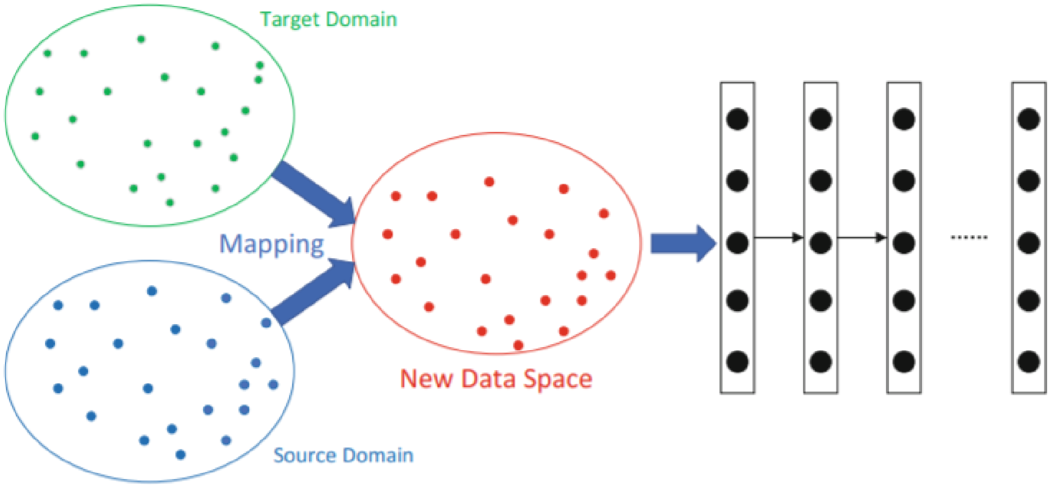

FIGURE 2.2: Transfer learning [1]

With the headways of Multi-access Edge Computing (MEC) and 5G advancements, modern applications are developing with unparalleled force and ongoing needs[32]. With regards to such executions, the prerequisite of AI (ML) procedures is to fabricate esteem from start to finish information. Current ML systems transfer information from geo-conveyed streams to a focal information model and the model is then moved to the edges and utilized for tendency or division. These systems may not work since they bring an incredible requirement for information stream and model transfer to a basic learning measure. In addition, a complete model may not be required in each separate area [33]. Figure 2.3 depicts the IoT devices also known as end devices like, cameras, mobile phones and printers, networked with the edge devices and the cloud services via embedded systems [2]. With the dangerous blast of shrewd devices and the appearance of many new bundles, site guests volume has been growing dramatically. We acknowledge the transfer study to refresh the prepared model (fundamental model) with a little part of the designated information to work on the precision of the anticipated speed model [34]. Coordinated conventional public structures can't oblige the singular's necessities because of the substantial burden on transport interfaces and significant distances. In this manner, new offices, which convey network abilities and content to the public area, are proposed, including PC registering and capacity. Portable component networks give distributed computing abilities and reserve stockpiling on the edges of versatile organizations [35]. In this study, we manage a comprehensive review of art research efforts in

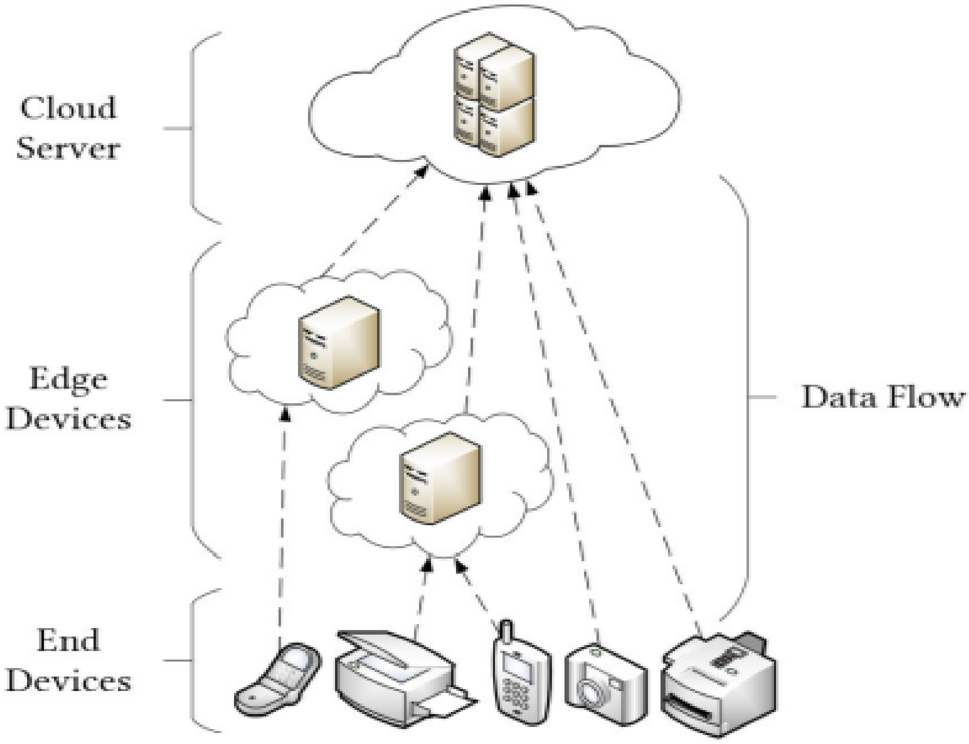

Figure 2.3: Data Flow over Edge computing [2]

mobile networks. We provide a framework for mobile side networks, such as definition, structure, and benefits. Next, a comprehensive survey of computer problems, caches, and communication strategies in the network [36] was conducted. At a medical services office, this will take into account the most basic prospects of clinical observing arrangements that will be straightforwardly connected to the medical services supplier, where patient subtleties will be investigated. Preferably, a model embedded in neighborhood information can be divided between suppliers to profit from information that isn't accessible locally, either to make a debias model or to acquire a superior comprehension of the surprising circumstance. Since the privacy of patient information is essential, we need a way to transfer data from sources that contain interest data without disclosing or sharing location data.

[37] author propose such an approach in terms of a drink of information that can transfer data between models in real-time. Our approach requires less training and more flexibility than the ability for various purposes that are rare or have never been well-recognized in the area. Joined with propels in profound learning (DL), this opens up unlimited freedoms for fundamental bundles, e.g., for helpful purposes and in the auto organization. Customary cycles dependent on provincial cloud (ML) necessitate that measurements be put away in a solitary area on a cloud worker or in the focal point

of realities. In any case, this has difficult issues identified with unsatisfactory postponements and shortcomings in word trade [38]. Accordingly, portable edge Computing (MEC) has been proposed to transfer the impediments as far as possible, where data is created. However, the standard technology for allowing ML on cell element networks still requires sharing non-public facts about external events, e.g., [26] promotes the most confidential mathematical laws and promotes security concerns; the idea of Federated of (FL) data obtained has been added. In FL, drop-off devices utilize their estimated realities to prepare the kind of ML needed by the worker. Devices give and send variant updates rather than green realities to the worker mix. FL can serve as a favorable generation in cell facet networks because it enables co-learning of the ML model and allows DL to optimize cellular networks. However, in an increasingly sophisticated and complex social society, unconventional gadgets with various obstacles are concerned [39]. This amplifies the difficulties of visit charging, the assignment of suitable assets, and protection and security inside the execution of the FL scale. In this survey, we start with a show on the FL legacy and qualities.
An intelligent decision analytical system needs a combination of decision-making and forecasting. The greater part of the assembling units depend on the knowledge base and request expectation of deals designs. Some of the sales data samples are not sufficient for the training of the system model that creates a negative effect on the forecasting. To tackle this issue, transfer learning can be used as it fits models that with a small set of data samples and manages the transfer of learning [40].

The following table summarizes the stated literature works. It depicts the enlightenment of each paper over transfer learning, machine learning, edge networks, and negative transfer learning. We denoted the research papers as the publication year, and the name of author and No represents no, and Y shows yes. The fields are saying No and Y state that the respective paper has discussed the facts of mentioned features or not, respectively.

Summary of Literature Review				
Research Papers	Transfer Learning	Machine Learning	Edge Networks	Negative TL
tsoumakas2019survey [9]	No	Y	No	No
chen2019data [10]	Y	Y	No	No
cheriyan2018intelligent [11]	No	Y	No	No
vzliobaite2009towards [12]	No	Y	No	No
jiang2019multi [13]	No	Y	No	No
eckart2021brief [14]	Y	Y	No	No
mamdouh2018securing [15]	No	Y	No	No
lee2012comparative [16]	No	Y	No	No
box2015time [17]	No	Y	No	No
rojas2013neural [18]	No	Y	No	No
day2017survey [19]	Y	No	No	No
afridi2018automated [20]	Y	No	No	Y
li2020baseline [21]	Y	No	No	Y
gopalakrishnan2017deep [22]	Y	Y	No	No
hou2018proactive [23]	Y	No	Y	Y
rosenstein2005transfer [24]	Y	No	Y	No
mohammed2021energy [25]	Y	No	Y	No
yang2020distributed [27]	Y	Y	Y	No
chen2019deep [28]	Y	Y	No	No
jiang2017transfer [29]	Y	No	Y	No
lim2020federated [26]	Y	Y	Y	No
li2018learning [31]	Y	No	No	No
khamparia2020internet [41]	Y	Y	Y	No
daga2019cartel [33]	Y	No	Y	No
wang2017survey [36]	Y	No	Y	No
goldstein2021decentralized [37]	Y	No	Y	No
lu2021adaptive [38]	No	Y	Y	No
zamir2018taskonomy [39]	Y	Y	No	No
wan2021review [40]	Y	No	Y	No

Chapter 3

Methodology

3.1 Introduction

This section focuses on the methodology and workflow of our study. Nowadays, numerous real-life applications are working successfully because of machine learning techniques; there are two fundamental research areas, i.e., classification and regression. This study focuses on these two applications; for the classification problem, we use heart attack detection as case study 1. For the regression problem, we use the mushroom demand prediction domain as case study 2. Transfer learning is a proven way to increase model efficiency within less time than traditional machine learning approaches.

Moreover, digging down in transfer learning, we explore the homogeneous transfer learning approaches for binary classification and prediction tasks. Figure 3.1 depicts the flow chart of the decisions and processes used in our methodology for obtaining optimistic results for the binary classification domain. Figure 3.3 shows the example of classification technique, we run five different machine learning algorithms over case study 1 (heart attack detection) and choose which one fits the best according to the accuracy score as shown in 4. And, we apply the feature-based homogeneous transfer learning approach using the source domain as a breast cancer detection system and calculated the accuracy score and the computation time as we implement the model on the edge device. To improve computation time, we use edge networks as the combination of embedded systems and edge devices.

Figure 3.2 represents the workflow of methodology for regression/prediction-based fields. We use the mushroom demand prediction dataset shown in Figure 3.4. Hence, it contains the periodic and seasonal data; we use the LSTM (long short-term memory) algorithm as

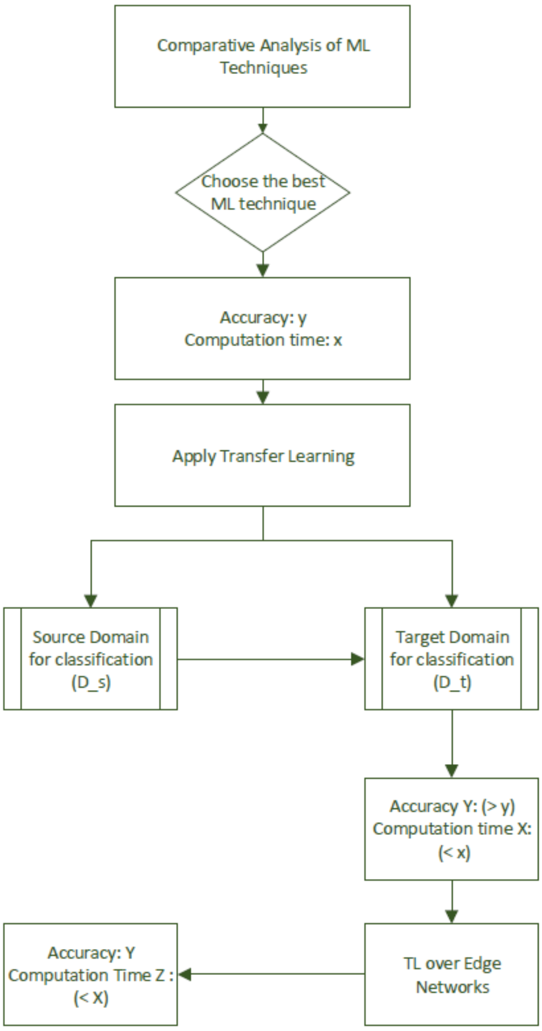

Figure 3.1: Using Transfer Learning for Classification based Application

a traditional ML model and compute the mean squared errors over prediction. Moreover, we applied the feature-based transfer learning approach to increase the model's efficiency and decrease the mean squared error. Still, as per the results shown in 4, it increased the MSE score of the model that results in the negative transfer. To avoid that, we use the hybrid-based transfer learning approach, a homogeneous technique, and evaluate the results and the computation time.

3.2 Application Dependent Transfer Learning

This section enlightens the knowledge of transfer learning from the organizations prepared for massive scope datasets to small datasets. We use the homogeneous transfer learning approach over the two case studies and propose a methodology to avoid the negative transfer happening using the existing transfer learning approach.

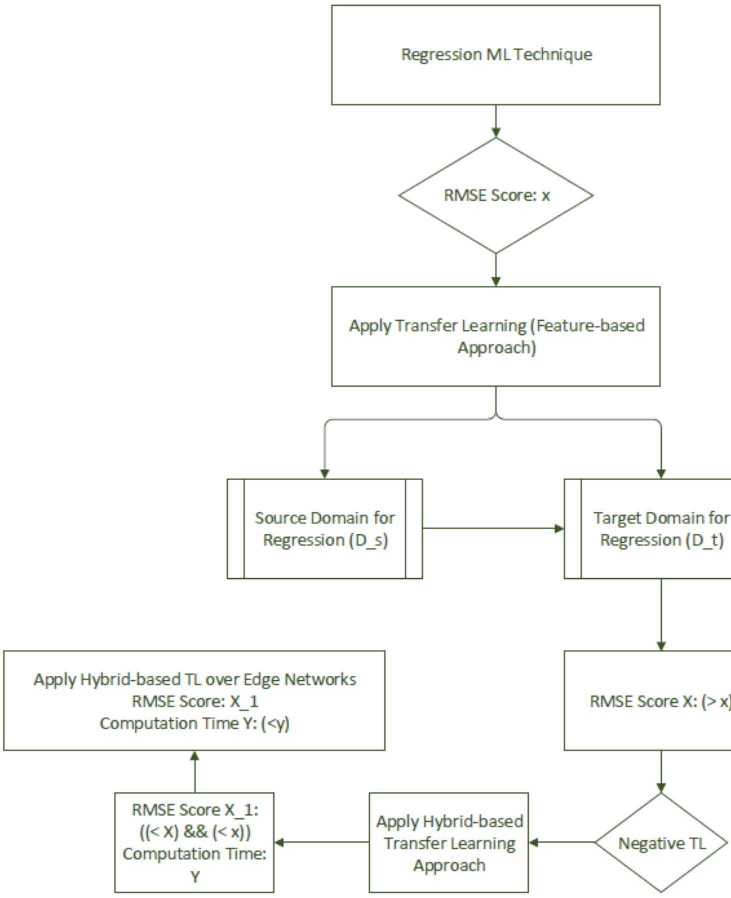

FIGURE 3.2: Using Transfer Learning for Regression based Application

Every transfer is domain-dependent; the target domain and the source domain have to have some relation. It can be the same number of attributes, similar distributes of the data samples or problems like classification or regression. We use the pre-trained models for both the domains and applied homogeneous transfer learning approaches. To make computer-assisted reading of calculation effective, convolution uses three key concepts, representing equivariant, parameters and sharing, and remote connections [42]. A small reference to CNN means that a few outputs of the current layer are connected to the next layer. In contrast, in other neural networks, all the production of the present layer neuron is connected to the Whole neuron input of the next layer. Since the covered area of each character (field reception area) is small, this continuously learns important features and dramatically reduces the calculated number of weights, which also improves algorithm performance. CNN minimizes the need for memory storage by using each kernel with its consequences for various areas of the complete image, known as weight sharing. In fully connected neural networks, central to the layers, are used once and then discarded. Weight sharing increases the quality of the static representation, which means that the

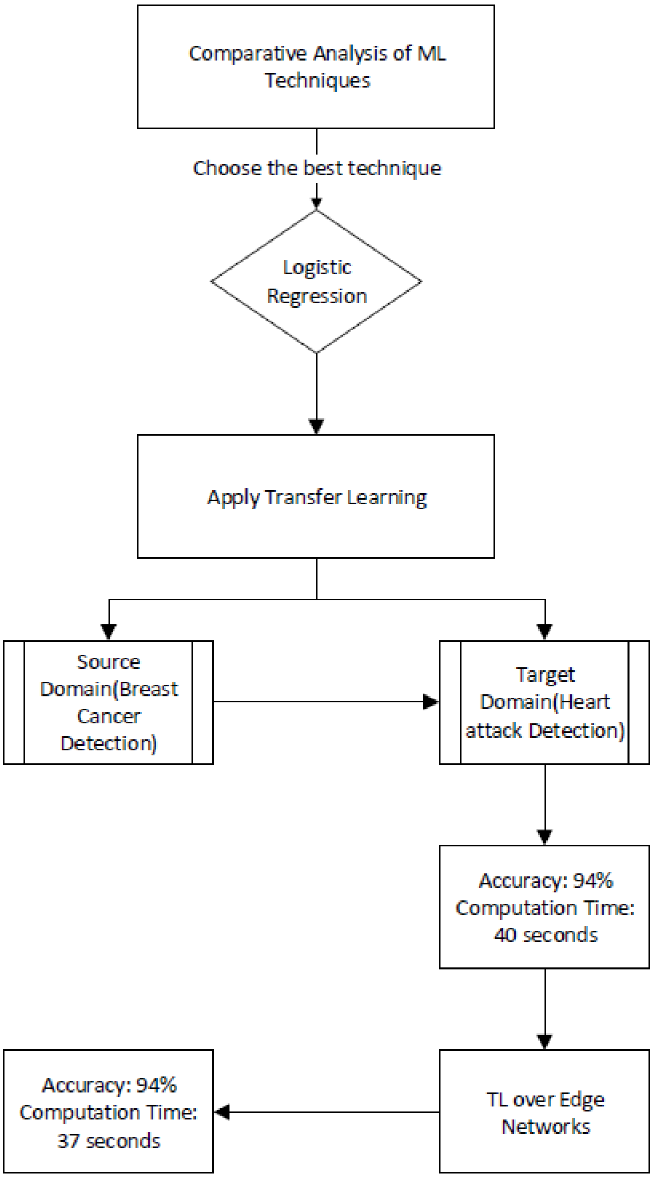

FIGURE 3.3: Example of Applying TL for Classification based Application

input translation leads to an equal translation of the feature map.

We use the breast cancer prediction system as the pre-trained base source domain for the target domain, the heart attack detection model. Moreover, we use the avocado price prediction system as the source domain for the regression problem of mushroom demand. We present a way to estimate the apparent analogies amongst two fields and picks a subset from the source area given the objective space. ImageNet-prepared Convolutional Neural Networks (CNN) have been madly used to communicate learning by utilizing a pre-prepared model as a component identifier or organization enhancement. According to the extraordinary accomplishment of utilizing CNN's recently prepared in transfer learning, huge endeavors have been made in understanding transfer learning. In

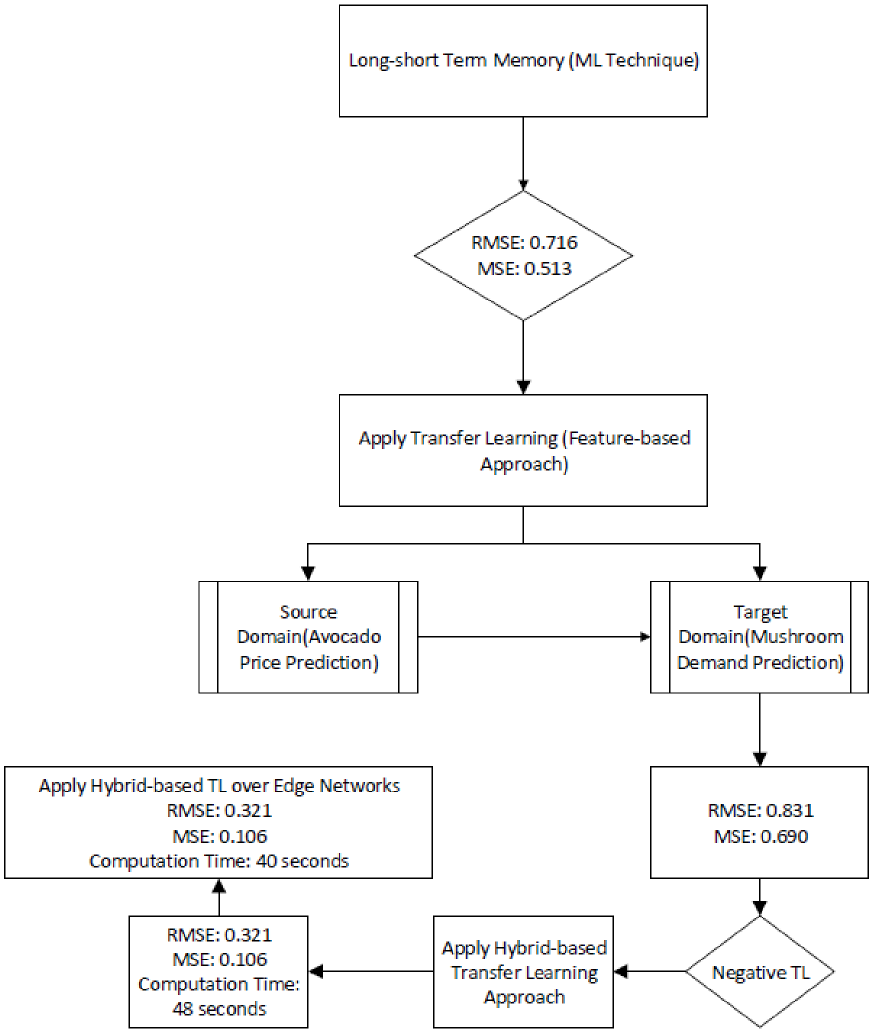

FIGURE 3.4: Example of Applying TL for Regression based Application

particular, some previous work has freely shown the link between transfer learning and domain similarity. For instance, transferring knowledge among contingent splits is more effortless than cracking a natural/manufactured object at ImageNet. Hence, manually integrating 512 appropriate categories from all available classes enhances ImageNet's widely used types in PASCAL VOC; transfers from the incorporated dataset of ImageNet and Places produce better outcomes in the rundown of visual capacities. Azizpour et al. have made a viable report in the rundown of move learning undertakings that have an alternate closeness to the genuine picture capacity of ImageNet (e.g., image fragment is considered the most similar to retrieval pattern, etc.). Our main difference between their work is twofold: First, we give an approach to gauge the likenesses between an area and a space source and select an equivalent setting from the source space for better exchange learning. Second, they utilized pre-prepared CNNs as trait coders and just

prepared the last layer or used SVM straightforwardly on the extricated provisions, and we calibrate all organization layers [43].
Data collection is the base of any fundamental model; the more critical data samples we collect, the system will be more enhanced because there will be enough training information to get precise outcomes. Moreover, the type of issue to be resolved is an additional main parameter affecting the system's efficiency. There may be a variety of system problem statements included as Regression problems that often require equal matrix measurement as a square or a division function that provides for clarification, recall, and f1 score[1].

We will learn the fundamental architecture of the transfer learning with respect to some high-level features. The clarified calculation Figure 3.5 shows the basic structure of the model.

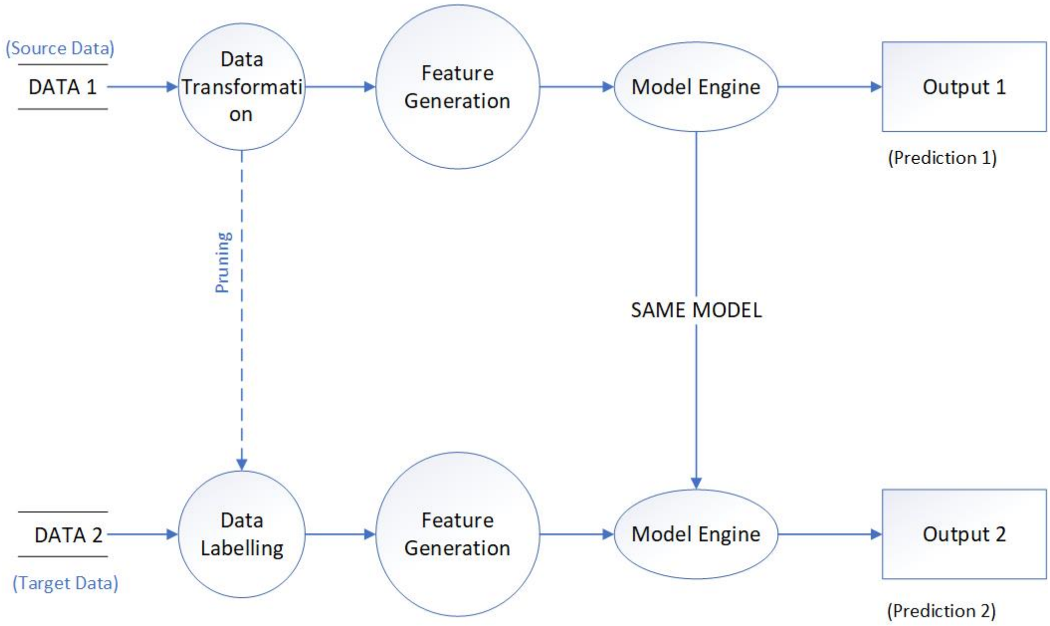

Figure 3.5: Basic Architecture of Transfer learning

It depicts when making any system; the most basic advance is to gander at the information factors and anticipated results and afterward take a gander at the availability of information tests. Our outcomes can be unsurprising when given data, and our available information focuses ought to cooperate adequately to provide a connection among sources of info and results. In this way, the AI of embedded systems is utilized to store practices from preparing information and is expressed in test information.

- Relatedness: Similarities often produce similarities based on prior knowledge. In this study, the relationship emphasizing finding connections among two data sets, i.e., source data and target. We can likewise say that specific cycles are engaged with the likeness search to make informational indexes interpret a similar dissemination, and methods are portrayed in the accompanying segments.

- Layer Freezing: This procedure is also known as the edge layer, which means that the tools of the training system do not change when we reuse it for the following function and remain stable. For example, the CNN model includes several layers instructed under large image databases; one can utilize it again by deducting the last layers and supplanting them with new layers. Moreover, those layers stay immaculate, and we can apply the above layers by applying a few changes utilizing the magnificent format.

- Pruning of data: As with transfer learning, users take a comprehensive network trained over various associated problems, and we process that as a fresh start to our personal/target work. For every neural network, circumcision is used to minimize stress. Our thesis uses this state of the art to determine the several; symbols in the database by order of approximate size in the source database, resulting in a much faster model.

In addition, when setting a goal, it is essential to set up an assessment process to approach that goal. The database should be divided into model training and testing. This move will prepare the system with a snippet of data, fix its factors utilizing approval, and test the test set's exhibition. The most common and efficient way to verify embedded programs is K-Fold authentication, which divides equal data sizes. In contrast to each phase, the system is instructed over K-1 cells.
Before training the framework of the system, there may be a requirement to process the data in advance. It is possible with continuous data that there may be few sparse parameters that cause problems with other algorithms, so we need to alter them prior to entering information straightforwardly into our models. Typically, lost qualities are signified as a "Invalid" marker. To resolve this issue, any components with stowed away factors can be taken out from the information base and converted unknown values into values that mean other features. Figure 3.6 represents a slope for improvement in

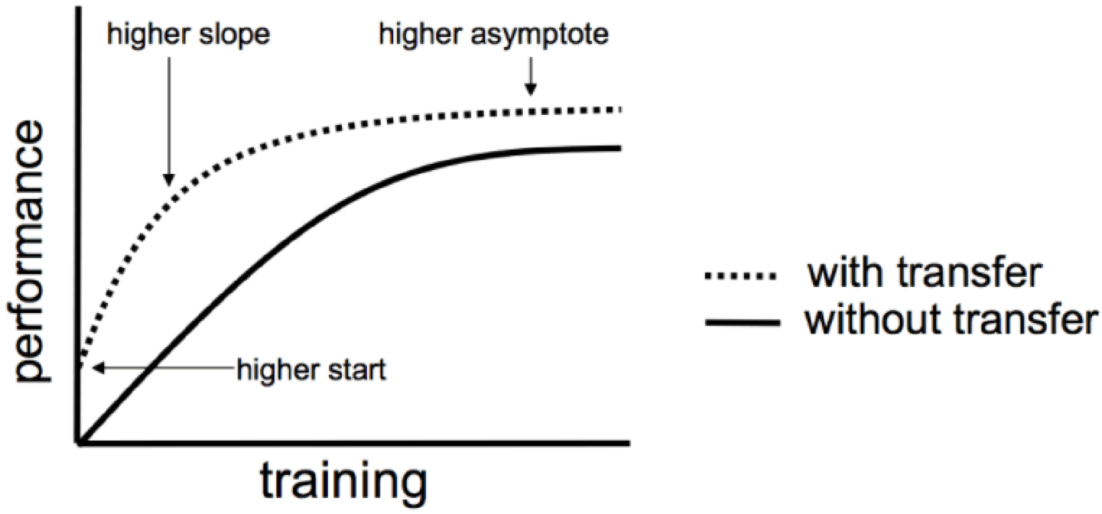

Figure 3.6: Improvement Slope

transfer learning. High start means the ability to start in the source model is higher than it would otherwise be. The higher slope indicates the level of skills developed during the training of the source model has increased more than it would have been. The high asymptote shows the training model after changing skills works better than the previous one.

It is essential to look at logical features to intercept over-modeling of the system that minimizes the number of useless items from the database. Heat maps can be created to explore relationships between all aspects of the data and eliminate those variables that show high (good) and low (bad) relationships.

The database can be divided into three stages of training, validation, and testing. We instruct the model with training details, check the advancement on the verification set, and, lastly, test it in the test database. For embedded systems, it is pivotal to have a validation stage as it tunes the arrangement and supplies input. The affirmation segment likewise affirms that the model can chip away at imperceptible components and foresee exact outcomes from new information. Information tests can be remembered for the normal outcomes model.

When a pre-trained source domain model is developed, this model is used as the primary building site. The model can incorporate all or a piece of it, contingent upon the system interaction. There are a few layers inside the structure of any neural organization model, which indicate the change of each picture to produce sufficient and exact outcomes. While making an exchange learning model, we need to dig further into those layers and apply them as per another space. But stated manipulations are integral to any transfer learning process to get the best outcomes for discrete yet associated fields.

Algorithm 1 Positive Transfer over Negative Transfer Algorithm

1: $\{f^{T_j}, f^{S_i}\}$ = Functions for feature-based transfer learning approach
2: $\{f_h^{T_j}, f_h^{S_i}\}$ = Functions for hybrid-based transfer learning approach
3: Test Module: Positive Transfer
4: **while** $\{(X^S, Y^S) \neq (X^T, Y^T) \mid (P(Y_S \mid X_S)) \neq (P(Y_T \mid X_T))\}$; **do** //For two different domains, or, tasks, they can have different feature space, label space, or different marginal probability distributions.
5: Process: $f^{T_j}(j = 1, ...t) = f^{S_i}(i = 1, ..., t)$; //implemented over 2.1.1.2 and 2.1.2.1.
6: Process1: $f_h^{T_j}(j = 1, ...t) = f_h^{S_i}(i = 1, ..., t)$; //implemented over over 2.1.1.2 and 2.1.2.1.
7: **if** (Transfer = 1) **then** //if transfer is positive
8: Implement Function '5'
9: **else**//If transfer is negative(Transfer = 0)
10: Implement Function '6'
11: **end if**
12: **end while**

Negative transfer is a result of not having expected output using a specific transfer learning approach. The above algorithm 1 depicts the process of avoiding the negative transfer in particular applications. This algorithm has two sets of functions, $\{f^{T_j}, f^{S_i}\}$ shows the working for feature-based transfer learning approach and $\{f_h^{T_j}, f_h^{S_i}\}$ depicts the process of hybrid-based transfer learning approach. Suppose both the domains' feature and label spaces are different, or the probability distribution is not the same. In that case, the user should apply the general homogeneous transfer learning approach; if the transfer learning is successfully implemented, i.e., the transfer is positive, the user should stick with the implementation. Otherwise, the user should implement a hybrid-based transfer learning approach, i.e., the user should focus on more than one technique and combine them to create a hybrid transfer learning approach as per the data requirements. The results using this algorithm are elaborated in section4.

We will use the Long short-term memory model (LSTM) and logistic regression for two case studies, i.e., mushroom demand prediction and heart attack detection, respectively. LSTM model anticipates the measure of the variable based on another feature's value. In our scenario, the mushroom sale depends on various factors like price, date, and type of mushrooms. On the other hand, logistic regression generally anticipates the binary dependant variable; this is a statistical model and uses a logistic function for binary predictions. Therefore, for heart attack prediction, we will have the outputs in a boolean variable. Moreover, we will read and examine the dataset for both case studies and remove the features with null or zero values. We are making our dataset free from sparsity problems, as this might affect the model's performance. Secondly, we will generate a co-relation map that is crucial before applying any machine learning algorithm. Co-relation map/ Heat map generates the value for all the dependencies in the dataset. It provides a better vision of essential and less critical independent variables, and we can further clean the data if needed.

The source and targeted domain data have many variables, so traditional machine learning methods are unsuitable for error categorization. The process adopted in this study decreases the cooperation between the source area and the objective space with an immediate transformation line. It limits the discrete information circulation among the source space, and the objective areas [44]. From that point onward, the vector objects of the covariance-related source space and the designated area are embedded into the SVM calculation for preparing and testing. The experiments show that the combination of the proposed covariance (COVAL) of error elements has higher accuracy in multi-country classification under more favorable operating conditions compared to other methods [5].

Transfer Learning focus on using broadly labeled data samples in the source domain to solve a unique yet related task, even if there are no similarities between the data problem of the training and testing and distribution features. Using transfer learning, training devices, or samples are required only to analyze the pre-trained model properly [45]. Figure3.7 shows the primary processing of transfer learning, such as building any model or partition. The main thing is to take a gander at the information sources and anticipated results and afterward take a gander at the accessibility of information. Our outcomes can be unsurprising when given data, and our accessible information ought to interface adequately to give a connection among data sources and results. Therefore, embedded machine learning systems store training data behaviors and transfer them to data testing. Data Collection is the premise of any learning model; the more supportive data we gather, the better we can make a model. There will be sufficient preparing information to get precise outcomes. Pushing ahead on the sort of issue to be addressed is another central point influencing the model's exhibition. There might be an assortment of embedded issue articulations for issues like Regression issues that often require a measurement error that means a matrix or a separation function that involves clarification, recall, and accuracy[46].

A co-relation map helps in enhancing the transfer learning technique as data cleaning is required while performing the knowledge transfer amongst two different domains. The components have to be similar to fit the target domain features into the pre-trained model on the source domain.

Hence, we will be using two case studies; for the first scenario, our source domain will be avocado price prediction, and the target domain will be mushroom demand prediction. Whereas for the other one, the source domain will be breast cancer prediction, and the target will be heart attack detection.
We are using mushroom sales prediction. This is a periodic dataset collected from the actual farm in Ontario, Canada; the data set contains data for only six months, which is insufficient to train the model and get accurate results. The source domain used for this case study is the avocado price prediction that has a comparatively more extensive dataset. We are using heart attack detection for the second scenario as this domain also has fewer data samples available and has some sparsity problems. We have used the breast cancer prediction domain as the pre-trained model with the more extensive data samples available to overcome this issue.

3.3 Transfer Learning over Edge Networks

Embedded systems contain functional and non-functional objects with various behavioral features that are run by several events. Behavioral elements manage events based on techniques acquired by the program developer. Central registering systems and models are mostly coordinated, have information designs put away and coordinated on edge devices, and send that information during handling. The basics of having data stored nearby should increase the ability to calculate.
Critical databases must be maintained and simultaneously required for transfer learning models while training, validating, and evaluating outputs. Ultimately, every TL model that works in any system can be costly, leading to a lack of flexibility and scalability. Applying the TL techniques over edge networks will be quicker than implementing them on the edge device.
Our proposed approach is to create a basic exchange learning model to take care of discontinuity and expectation issues, put them on the device edge, and assess the handling time. Furthermore, the next move is to recount the transfer learning model in the

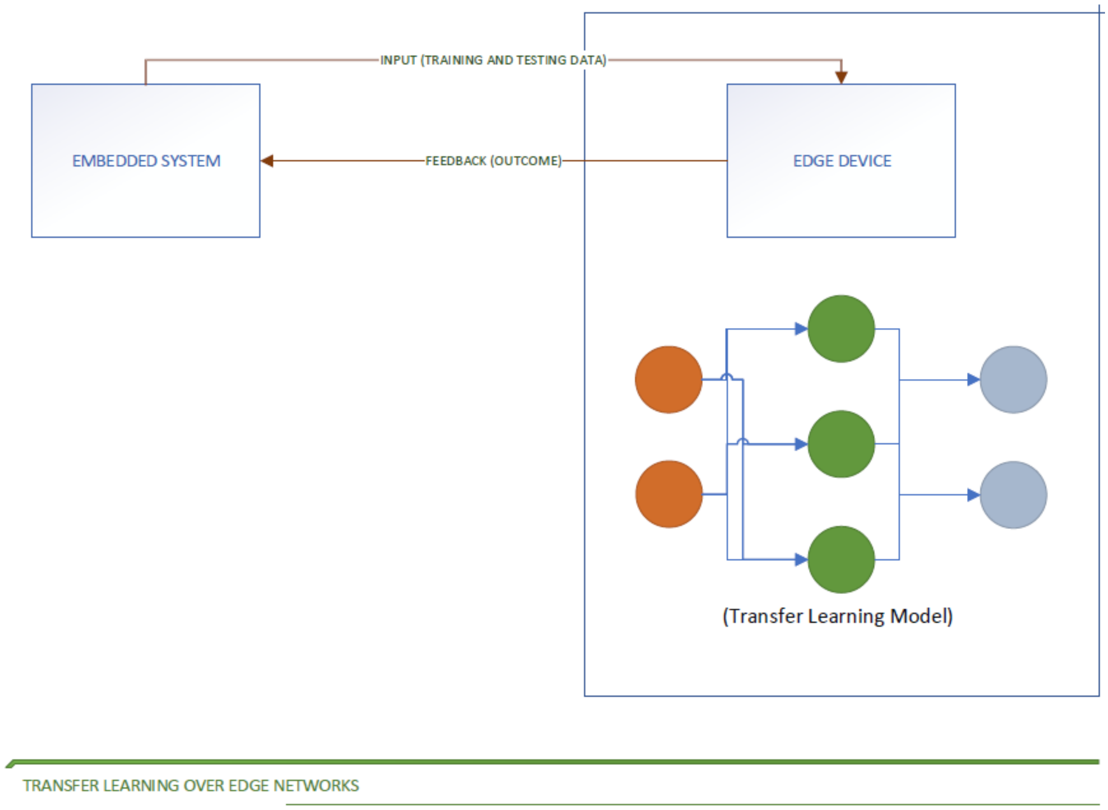

FIGURE 3.7: Transfer learning on Edge Networks

Algorithm 2 Transfer Learning on the Edge Networks [8]

Triggers : Request for process
Test Modules : Edge Device, Embedded System
while processRequest.exist(); **do** //Checking whether any requests are present or not
    edgeDevice: processTime_ED = requestProcessedTime()
    embeddedSystem: processTime_ES = requestSendTime() + requestProcessedTime() + responseSendTime()
    if processTime_ED > processTime_ES **then**
        'ES'
    else
        'ED'
    end if
end while

embedded system and organize it with the edge device to get to information, otherwise called edge figuring. From that point onward, assess the arrangement time and pinging time stamps that send the reaction and get the response between the edge and the embedded system.

TL technique can be implemented over local devices and embedded systems. Our tests can work very well and without network delays when performed on an embedded system. Additionally, the calculation portrays which technique will work best to utilize the exchange learning framework. In Algorithm 2, it signifies the reasons for getting a solicitation for starting the methodology. To play out any occasion, it needs to be begun, and the application interaction is the start of the model and can be mentioned at whatever point important to execute the procedure. One more basic component referenced in the calculation is the test module addressing two frameworks that look at handling time. That is the edge gadgets and implanted frameworks. In the event that any applications utilizing the gadget model anxious, time would be a period for direct activity. In embedded systems, the entire processing time will be delivery time, action time, response time. The embedded system emphasizes executing a particular function of using a transfer learning model better than a separate system.

Chapter 4

Experimental results and analysis

We will be performing our experiments on two case studies, mushroom demand prediction, and heart attack detection. Firstly, we will apply the traditional machine learning techniques for heart attack detection problem, compare the accuracy results of the various ML algorithms. And implement the long short-term memory model (LSTM) on the first case study, i.e., mushroom demand prediction. Secondly, we will use a hybrid-based transfer learning approach on the first target domain. We will compare the accuracy of the performance results with the traditional machine learning approach that is LSTM. We have used the source domain as Avacado price prediction that contains periodic data and pre-trained on LSTM. Moreover, we will implement the Logistic Regression on the second case study as the source domain for that problem is breast cancer prediction that is also pre-trained over logistic regression and having an accuracy value of 93 percent. Also, we will be using the transfer learning techniques on the edge networks and compare the efficiency of the transfer learning approach on the embedded system. This section is further categorized into three subsections as follows:

4.1 Traditional Machine Learning

We will utilize a momentary model to recollect the estimate of mushroom interest. LSTM is RNN that can learn long haul reliance between succession information groupings. Dissimilar to CNN, LSTM can reflect network status during conjectures.

The fundamental provisions of the LSTM network are the succession input layer and the LSTM layer. The arrangement input layer embeds the time-series information into the organization, and the LSTM layer learns the drawn-out reliance between the grouping information ventures over the long run.

LSTM is prepared to isolate information and time, during which the forecast or yield of an organization ought to be founded on a retained succession of information focuses. LSTM has the same control flow as a recurring neural network. Processes data transmission as it spreads forward, and the difference is the performance within the LSTM cells. The focal idea of LSTM is the condition of the cell and with different entryways. The state of the cell goes about as a vehicle expressway that communicates data identified with all groupings. One can consider it the "memory" of an organization. The condition of the cell, in principle, can convey pertinent data all through the grouping [47]. So even data from past activities can make it a point for later occasions, lessening the impacts of short memory. As the situation with the cell proceeds with its excursion, information is added to or eliminated from the cell by the door. Doors, through various neural organizations, figure out which data is permitted in a cell. Doors can realize what data to keep or forget during preparation. We have applied this model over the data of mushroom farms as it is the periodic and seasonal data, and the independent variable in the data set is "date." The dependant variable is "sales." We have got the RMSE (root mean squared error) score of 0.716 and MSE (mean squared error) score of 0.513 as shown in the figure 4.1.

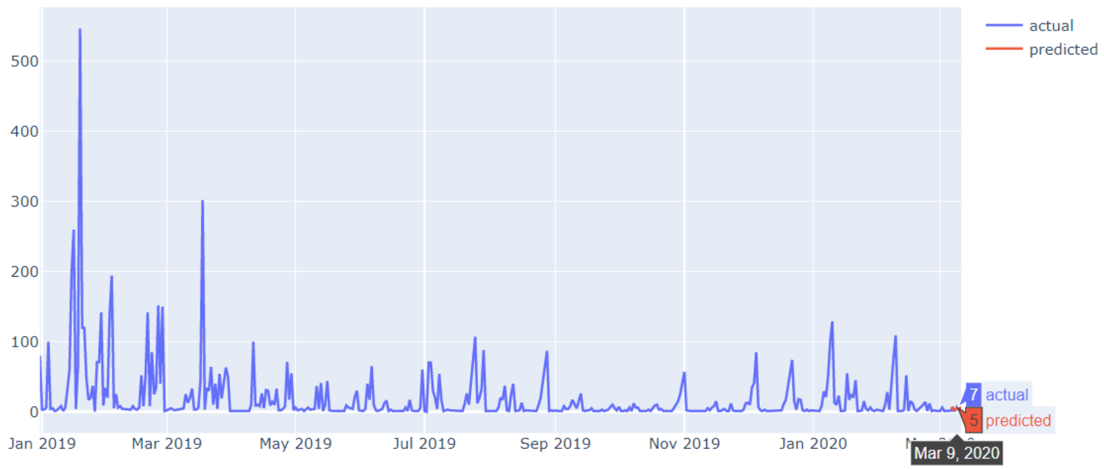

FIGURE 4.1: Mushroom Demand Prediction using LSTM

For the case study, heart attack detection system, we used five different classification machine learning algorithms and compared the performance using the classification chart, i.e., SVM, NB classifier, random forest classifier, decision tree classifier, and Logistic

Regression. The objective of the support vector machine technique is to detect a hyperplane in the corresponding space N (N - number of elements) that separates the data points. The support vector machine is highly preferred by many as it produces remarkable accuracy with low calculation power. The Support Vector Machine, abbreviated as SVM, can be used for retrofitting and split operations, and however, it is widely used for planning purposes.
In the SVM algorithm, we are looking to increase the boundary between data points and hyperplanes. The loss function that helps to increase the margin is to lose the hinges. Now that we have a loss function, we take the output in part in weight gain gradients. Using gradients, we can update our instruments. There is no separation, i.e., our model precisely predicts the period of our information point; we should refresh the angle from the standard boundary. Where there is an incorrect classification, i.e., our model makes an error in the prediction of our data point section, we include the loss and the standard parameter for performing the gradient update. As we implemented SVM on the heart attack detection system, we got an accuracy of 84 percent, as shown in the classification chart 4.2 below:

Support Vector Machine				
#	Precision	Recall	f1-score	Support
0	0.92	0.7	0.79	33
1	0.8	0.95	0.87	43
accuracy			0.84	76
macro avg	0.86	0.83	0.83	76
weighted avg	0.85	0.84	0.84	76

Figure 4.2: Classification Report of SVM

The Naive Bayes classifier is a method used to do the disengaging limit. The embodiment of division relies upon the possibility of the Bayes speculation. Using the Bayes speculation, we can find openings for A to occur considering how B has occurred. Here, B is confirmation, and A is the theory. The premise made here is that the assumptions/features are independent, that the presence of one part doesn't impact the other. In this manner, it is called Naive. Naive Bayes is the most untroublesome portrayal model. Using the Naive Bayes classifier, we got an accuracy of 82%, as shown in the going with gathering report.4.3.

Naivebayes Classifier				
#	Precision	Recall	f1-score	Support
0	0.86	0.73	0.79	33
1	0.81	0.91	0.86	43
accuracy			0.82	76
macro avg	0.83	0.82	0.82	76
weighted avg	0.83	0.83	0.83	76

FIGURE 4.3: Classification Report of NB

The Random forest, as its name implies, is home to many decision trees that serve as a single group. Each tree in a random forest spells out a category forecast, and a category with the most votes becomes a forecast for our model. Low integration between models is key. Just as low-correlated investments come together to form a portfolio more extensive than the sum of its components, uncomplicated models can produce more accurate forecasts than any other prediction. This positive effect is that the trees protect each other from their mistakes (as long as they do not always deviate in the same way). While some trees may be bad, many other trees will be good so that the trees can move in the right direction as a group. Random forest is a segregation algorithm with many tree choices. It uses random wrapping and installation in the construction of each tree in an attempt to create an unrelated forest of trees whose committee predictions are more accurate than any individual tree. Results of implementing random forest classifier on heart attack detection problem are shown below in 4.4

RandomForest Classifier				
#	Precision	Recall	f1-score	Support
0	0.77	0.82	0.79	33
1	0.85	0.81	0.83	43
accuracy			0.81	76
macro avg	0.81	0.82	0.81	76
weighted avg	0.82	0.82	0.82	76

FIGURE 4.4: Classification Report of Random Forest Classifier

The decision tree is a machine learning model that uses rules to make decisions like how people make decisions. Another way to think about this algorithm is that it is designed to make decisions. Decision trees can do sorting and reversing tasks, so you will see the authors refer to them as the CART algorithm: Classification and Regression Tree. This is an umbrella term, which applies to all tree-based algorithms, not just tree trunks. The feeling behind Virtual Reality is that you use the database features to create yes / no questions and further classify the database until you extract all the data points for each class. In this process, you organize the data into a tree structure. Every time you ask a question, you add a node to the tree. The first node is the root node. The query result separates the database based on the feature value and creates new nodes. If the user decides to stop the process after splitting, the last nodes created are decision trees. The following figure shows the classification and accuracy report for the heart attack detection system over decision tree classifier 4.5.

Decision Tree Classifier				
#	Precision	Recall	f1-score	Support
0	0.77	0.73	0.75	33
1	0.8	0.84	0.82	43
accuracy			0.79	76
macro avg	0.79	0.78	0.78	76
weighted avg	0.79	0.79	0.79	76

FIGURE 4.5: Classification Report of Decision Tree Classifier

We used the Logistic regression model to anticipate a heart attack in the second study. First, we need to produce a collaborative data map for data analysis. After creating the temperature maps, we set the boundaries using Equation. It is essential for pre-data processing due to abstract data, errors, inconsistencies, outliers, and a lack of variable values. To achieve practical analysis, pre-processing data strategies such as external detection, the cleaning method, modification, and data integration should be done before the merger process. Normalization is one more significant stage in pre-handling information to set the upsides of all factors from a variable scale to a specific scale range. Anomalies straightforwardly influence the exhibition of information examination.

$$x = \frac{x_{data} - np.min(x_{data})}{np.max(x_{data}) - np.min(x_{data})}.values \tag{4.1}$$

We then, at that point, split our train test information utilizing the "train-test-split" library at 80:20 [46]. After that preparation information, we began instruments and determinations by characterizing sigmoid () capacity and utilizing Equation 4.1. The sigmoid capacity is known as the sigmoidal bend or the section work. It contains a blend of limitless yield, and this is likewise an answer for the normal condition [48].

$$y_{head} = \frac{1}{1 + np.exp(-z)} \tag{4.2}$$

Strategic Regression is a straightforward neural organization that isolates information. For instance, grouping messages as spam or non-spam is the old utilization of recovery resources [49]. Resource recovery takes input, moves it through sigmoid capacity, and afterward returns the likelihood result somewhere in the range of 0 and 1. This sigmoid capacity is answerable for the information partition. And afterward, we start with the model boundaries. The following stage is to utilize a capacity considered appropriation that peruses the boundaries by adding the expense work (front) and its angle (back) [50].

Subsequent to acquiring an inclination for transmission and streaming in reverse, we amended all learning boundaries. We utilized Logistic Regression by characterizing strategic relapse work and utilized it multiple times.

We have reported the accessible information on bosom malignant growth and the foundation of cardiovascular failures. While the designated area, notwithstanding, has nine characters, however, it needs more example focuses. The accessibility of a marked informational collection assists with making the test more proficient. Figure 4.6 shows the accuracy score and the cost function over 510 iterations of heart attack detection system using logistic regression. Moreover, figure 4.7 depicts the classification report of the case study.

Figure 4.8 represents the bar graph for the accuracy score of five classification machine learning techniques as shown in the graph, the highest accuracy of 84% is obtained using a support vector machine, and the least accuracy of 79% is using decision tree classifier. As the graph shows, every algorithm provides the average accuracy falling in the range of 79% to 84% that can be claimed as a good accuracy for the model to work.

4.2 Application Specific Transfer learning

For the legitimate use of transfer learning strategies, we play out an illustrative review in medical services frameworks and the food business to demonstrate the ideas of transfer Learning among two distinct regions to take care of comparative kinds of issues, i.e., characterization and regression. The source domain for the order task is the Breast

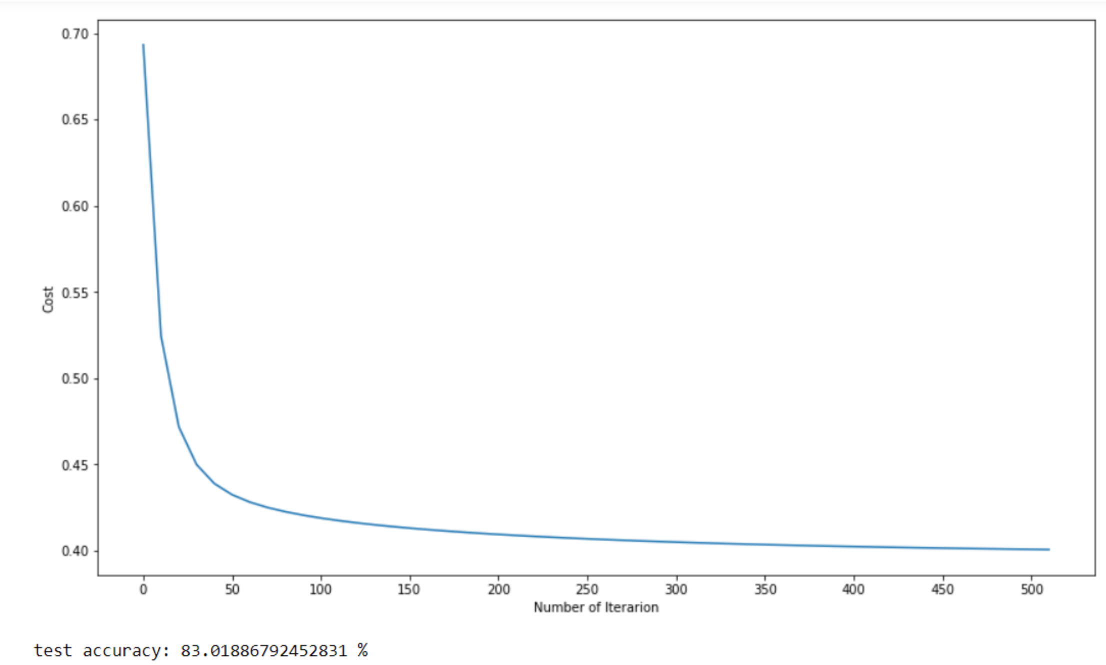

FIGURE 4.6: Heart attack prediction using Logistic regression

Logistic Regression				
#	**Precision**	**Recall**	**f1-score**	**Support**
0	0.86	0.73	0.79	33
1	0.81	0.91	0.86	43
accuracy			0.83	76
macro avg	0.83	0.82	0.82	76
weighted avg	0.83	0.83	0.83	76

FIGURE 4.7: Classification Report

disease discovery framework model, and the source area for relapse issue is Avocado value forecast; there are pre-prepared models accessible for the source areas that are prepared on the bigger informational collections and foreseeing better outcomes utilizing the best AI calculations, i.e., Logistic Regression and LSTM, separately. The objective space is the Heart assault forecast framework and Mushroom request expectation; we have more modest informational collections for these objective areas similarly.

The accompanying arrangement of informational indexes 4.9 is utilized to distinguish bosom malignant growth, which is our Task A with a lot of information, with six significant boundaries, to be specific, range, surface, edge, perfection, area, and analysis.

We utilize 80 % of preparing information and 20 % in testing for the two subjects. For the Heart attack detection problem, every one of the information in the source and target

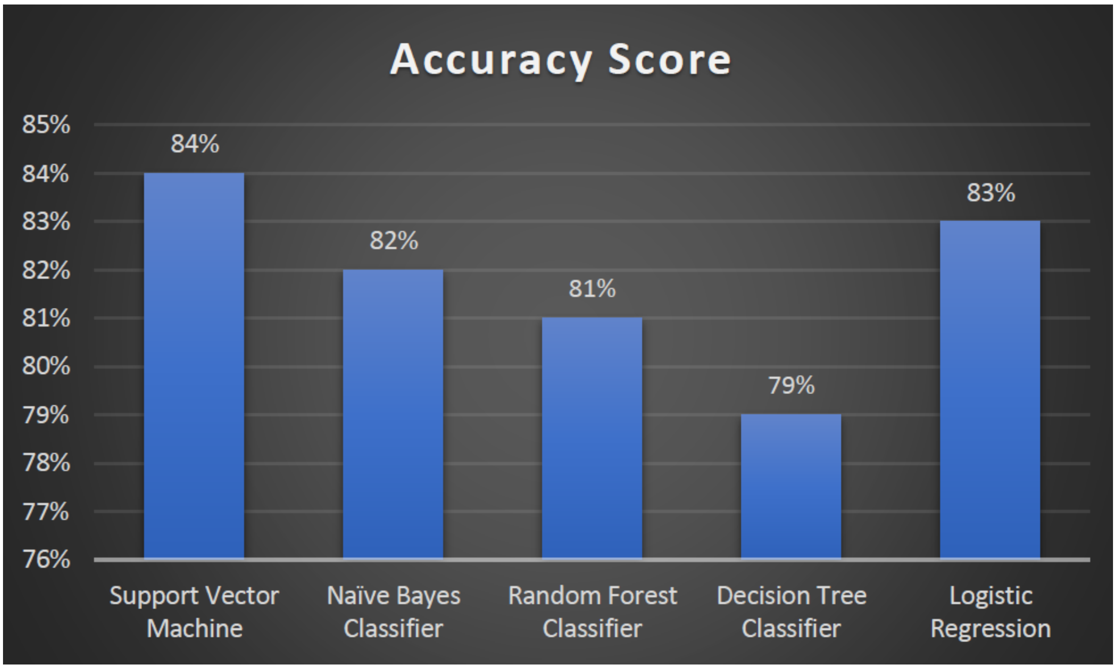

FIGURE 4.8: Comparative analysis of ML Models Over Heart attack detection system

area is in a numeric structure, which makes it simpler to perform Logistic Regression for the identification. For Mushroom demand prediction problem, we have periodic data of 6 months with dates in target domain and for the source domain, we have the data available for 24 months including time period as well.

mean_radius	mean_texture	mean_perimeter	mean_area	mean_smoothness	diagnosis
17.99	10.38	122.8	1001	0.1184	0
20.57	17.77	132.9	1326	0.08474	0
19.69	21.25	130	1203	0.1096	0
11.42	20.38	77.58	386.1	0.1425	0
20.29	14.34	135.1	1297	0.1003	0
12.45	15.7	82.57	477.1	0.1278	0
18.25	19.98	119.6	1040	0.09463	0
13.71	20.83	90.2	577.9	0.1189	0
13	21.82	87.5	519.8	0.1273	0
12.46	24.04	83.97	475.9	0.1186	0
16.02	23.24	102.7	797.8	0.08206	0
15.78	17.89	103.6	781	0.0971	0
19.17	24.8	132.4	1123	0.0974	0
15.85	23.95	103.7	782.7	0.08401	0
13.73	22.61	93.6	578.3	0.1131	0
14.54	27.54	96.73	658.8	0.1139	0
14.68	20.13	94.74	684.5	0.09867	0
16.13	20.68	108.1	798.8	0.117	0
19.81	22.15	130	1260	0.09831	0

FIGURE 4.9: Data-set for Breast Cancer detection system

Figure 4.9 depicts the dataset for source domain; breast cancer prediction (Task A). The Knowledge learned from Task A is transfer to the following data-set Fig 4.10 which is Heart Attack Detection, Task B. It has nine parameters: age, sex, Chest pain type, resting blood pressure, cholesterol, fasting blood sugar, Resting ECG, maximum heart rate achieved, old peak, exang, num [8].

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak
28	1	2	130	132	0	2	185	0	0
29	1	2	120	243	0	0	160	0	0
29	1	2	140	?	0	0	170	0	0
30	0	1	170	237	0	1	170	0	0
31	0	2	100	219	0	1	150	0	0
32	0	2	105	198	0	0	165	0	0
32	1	2	110	225	0	0	184	0	0
32	1	2	125	254	0	0	155	0	0
33	1	3	120	298	0	0	185	0	0
34	0	2	130	161	0	0	190	0	0
34	1	2	150	214	0	1	168	0	0
34	1	2	98	220	0	0	150	0	0
35	0	1	120	160	0	1	185	0	0
35	0	4	140	167	0	0	150	0	0
35	1	2	120	308	0	2	180	0	0
35	1	2	150	264	0	0	168	0	0
36	1	2	120	166	0	0	180	0	0
36	1	3	112	340	0	0	184	0	1

FIGURE 4.10: Data-set for Heart Attack detection system

It is important to run pre-processing of data because it contains much errors, inconsistencies, outliers, and volatile values. To obtain a successful analysis, several pre-processing data such as external acquisition, the cleaning method, modification, and data integration should be done before the merger process. Performing normalization is one more significant stage in pre-preparing information to set the upsides of all factors from a variable scale to a specific scale range. Exceptions explicitly influence the exhibition of information examination [51].

Every cell in the table shows the association between the two factors. The integration matrix summarizes data, such as input into highly advanced analysis, and as an advanced analysis tool. The purpose of editing an integration matrix is to summarize the large amount of data where the goal is to identify patterns. In our example above, the obvious way is that all variables are highly correlated with each other, and the inclusion of other analyzes. For instance, people often use aggregate matrix such as object-oriented analysis, validation of items, proof of models, and linear regression, where the missing values can be added in pairs. Moreover, as a diagnostic when looking at other analyzes. For example, a higher junction value with line rotation suggests that linear line estimates are unreliable.

The data we use to calculate the merge usually contains missing values. This could be because we did not collect this data or did not know the answers. Various strategies are available to deal with missing values when installing an integration matrix. The best practice is usually to use multiple imputations. However, people often use the lost values pairwise (sometimes known as partial combinations). This includes computer-integration using all available data of two variants. Alternatively, some use listed deletions, also known as intelligent deletions, which only use detection without lost data. Both paired and clever assumptions assume that data is completely lost at random.

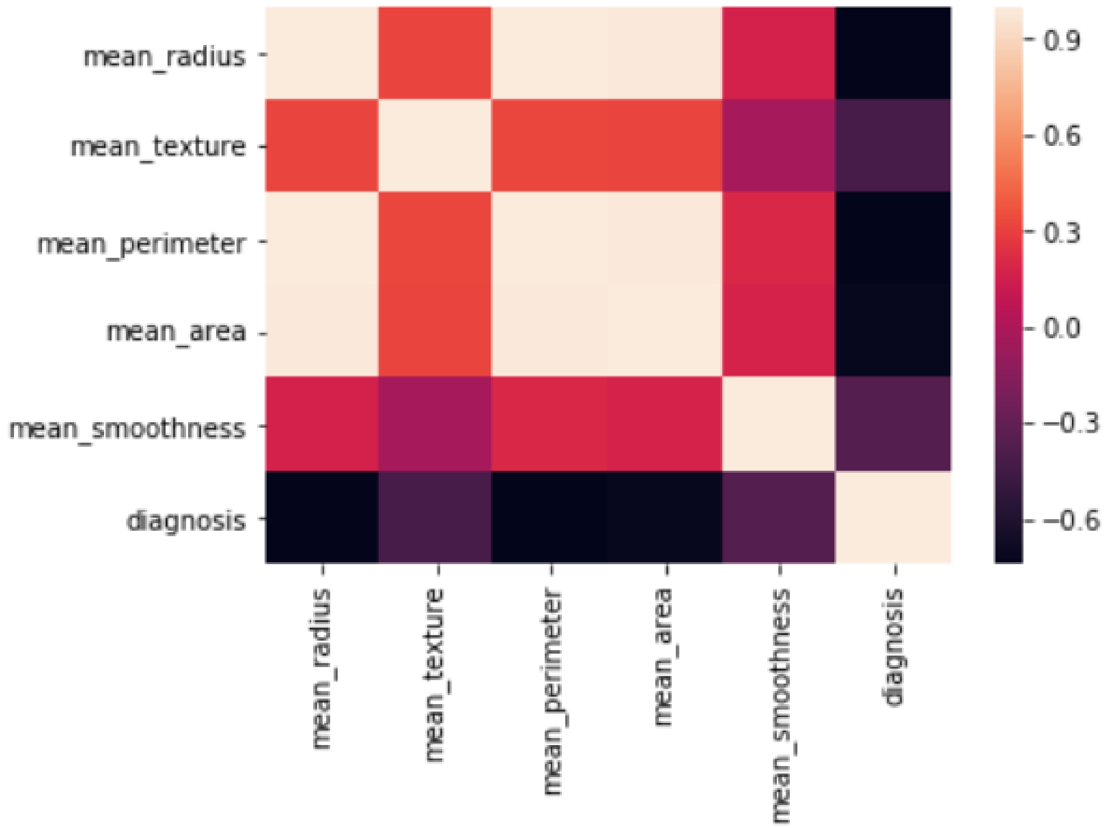

FIGURE 4.11: Co-relation map for breast cancer prediction

Calculated regression is a straightforward neural organization that isolates information. For instance, ordering messages as spam or non-spam is the old utilization of recovery resources. Returning a resource takes input, moves it through a capacity called sigmoid capacity, and afterward returns the possible outcome somewhere in the range of 0 and 1. This sigmoid capacity is answerable for the information partition. And afterward, we start with the model boundaries. The subsequent stage is to utilize a capacity called

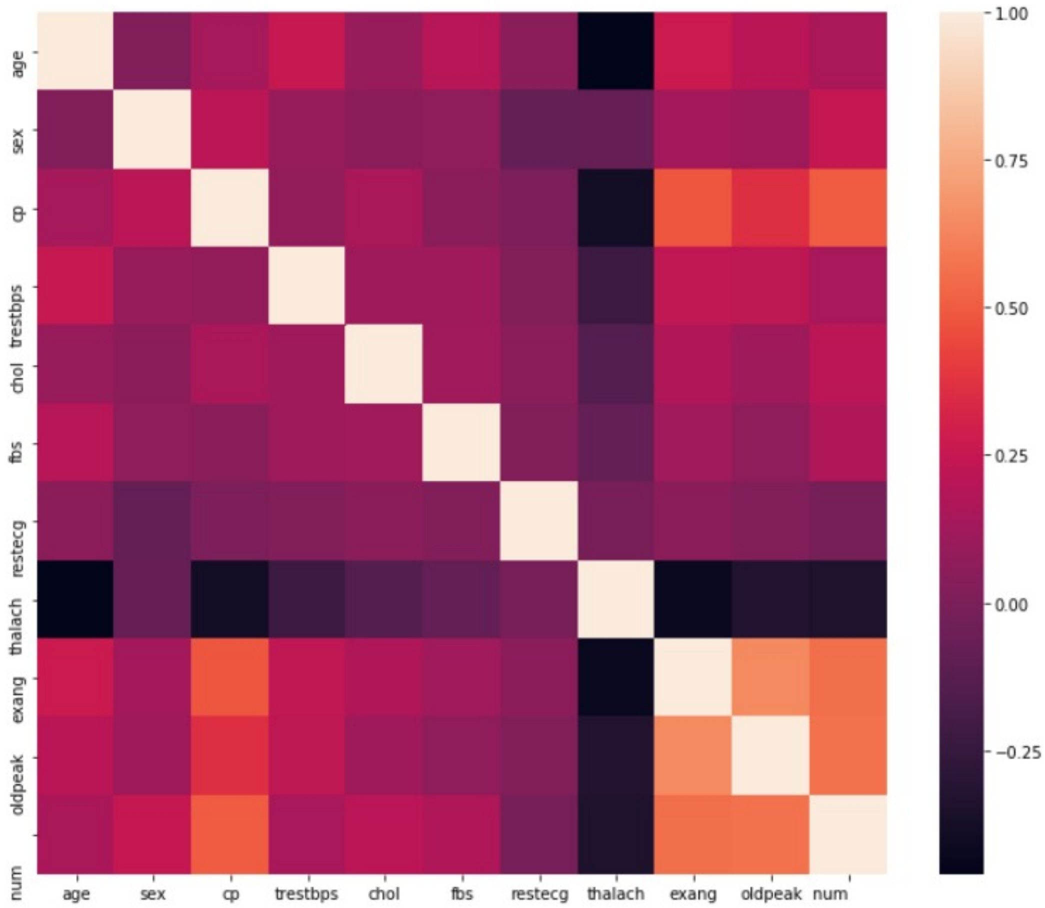

FIGURE 4.12: Co-relation map for heart attack detection

appropriation () that peruses the boundaries by entering the expense capacity and its inclination [50].

Subsequent to getting an inclination for transmission and streaming in reverse, we reconsidered all learning boundaries. We utilized Logistic Regression to characterize the strategic relapse () capacity and afterward performed it multiple times and discovered the expense after all duplication and to print the precision of the test which was 94% in our designated tasks, anticipating heart attack detection in Figure 4.13.

We have recorded the accessible information on breast cancer and the foundation of heart attack detection. While the designated space, in any case, has nine characters, however it needs more example focuses. The accessibility of a named informational index assists with making the test more productive.

Moreover, implementing the LSTM using a homogeneous feature-based transfer learning approach, we have the pre-trained system model for the avocado price prediction. We

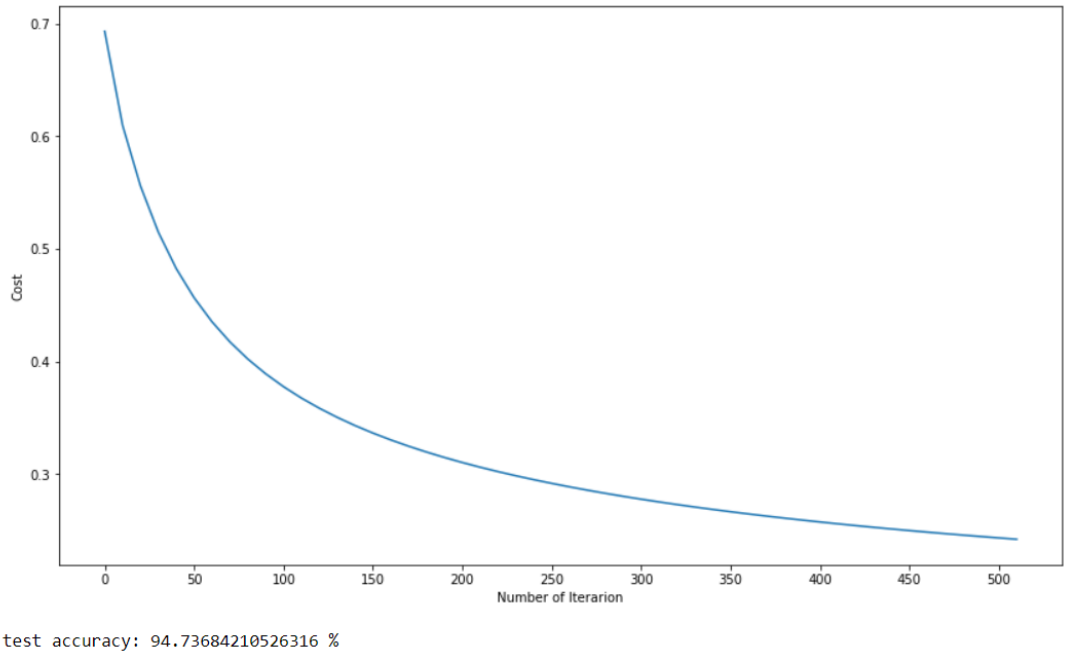

Figure 4.13: Heart Attack Prediction using TL

integrated the data from the source and target domains, i.e., mushroom demand prediction, into one domain to make sufficient training data samples. And tune the pre-trained model on the training data for the target domain. We first applied the feature-based approach as shown in the figure 4.14 gives the MSE score of 0.690 and RMSE score of 0.831. Figure 4.15 shows the results of compiling a hybrid-based transfer learning approach that provides the MSE score of 0.106 and RMSE score of 0.321 that is better than the results obtained using the feature-based approach and traditional machine learning approach.

We considered the feature-based approach showing the negative transfer as it is not providing the efficient results, MSE and RMSE score is more than the scores while compiling ML method and, after applying hybrid-based approach we obtain the better results. We can conclude from the results if the feature-based approach is not giving the expected outcome, then one should go for the hybrid-based technique.

4.3 Transfer Learning over Edge Networks

After using our TL model, we decide on the best plan to use our method. Our review pushes ahead to the edge organizations, moreover, in applying our model over edge networks, we utilize our nearby framework as an local system that plays out a few

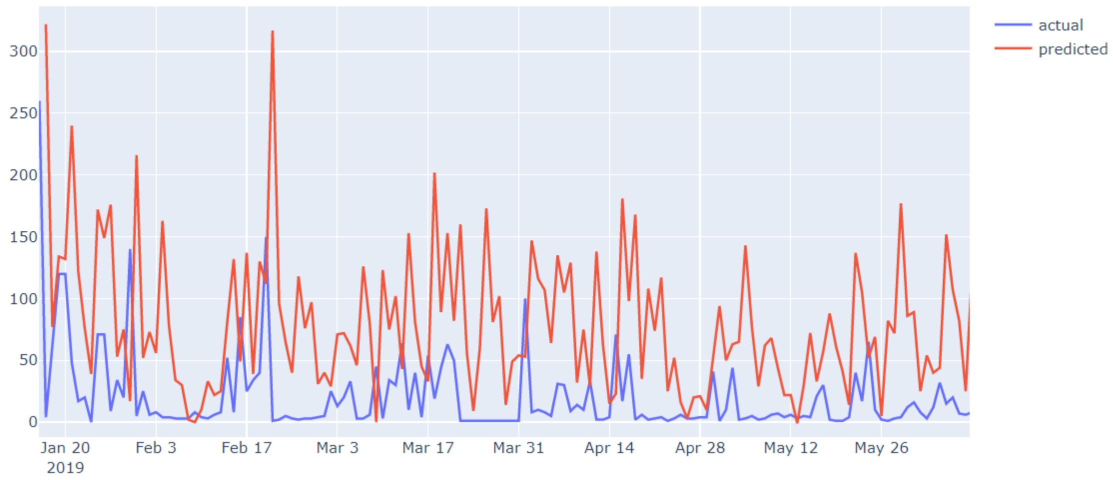

FIGURE 4.14: Transfer learning using feature-based approach for mushroom sales prediction

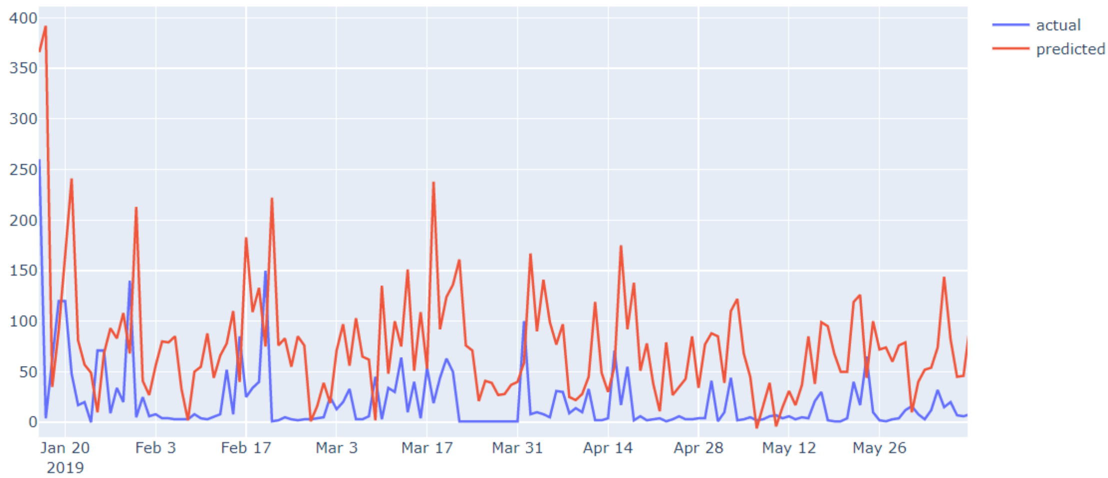

FIGURE 4.15: Transfer Learning using hybrid-based approach for mushroom sales prediction

assignments at the same time and has data sets put away on it. Another program is an installed framework with just the communicated perusing model that we made and acquired informational indexes by interfacing the embedded system to the edge.

All the experiments are done on the following systems;

- We use the laptop as edge device with Intel(R) Core i7-5600U CPU has 4 core processors and the RAM of 8GB with 256 solid state drive (SSD). It has x64-bit based operating system.

- We make a virtual machine to replicate the embedded system having 2 core processors, internal memory of 2GB and 64-bit operating system (ubuntu 20.04.01).

We use Anaconda and jupyter notebook as a platform to compile the program, and we are using various python libraries like pandas, matlplotlib.pilot, numpy, seaborn, TensorFlow for the implementations of the Logistic Regression and LSTM.

We implemented the techniques stated in Section 4.1 over edge device and embedded systems individually, and computed the time taken by each algorithm. The total time taken over edge networks, including compilation and pinging among systems for logistic regression, decision tree classifier, random forest classifier, naive Bayes classifier and SVM are 40, 77, 74, 76, and 82 seconds, respectively. Moreover, implementing the mentioned algorithms over local systems takes 67, 93, 80, 118, and 124 seconds, respectively. We compiled the machine learning algorithms for many times and calculated the average time to get the precise and efficient results. Figure 4.16 represents the graph and table for the better understanding of the results obtained.

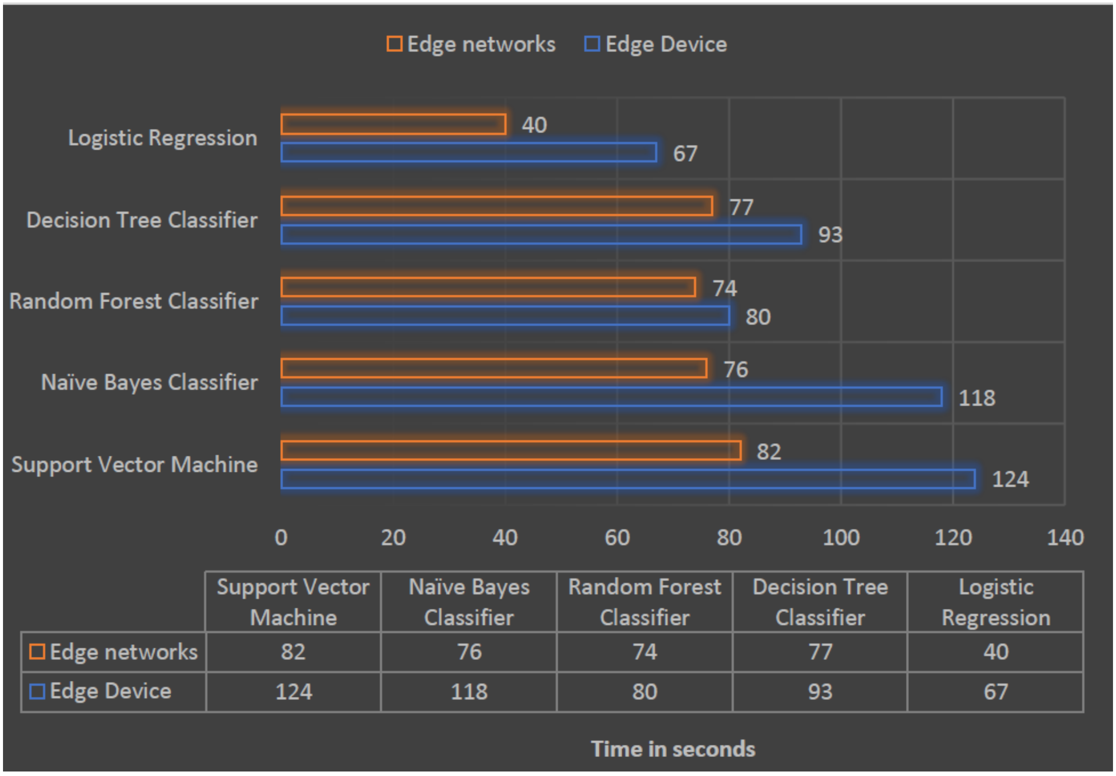

FIGURE 4.16: Computation time of ML techniques over Edge Networks

Since edge networks works well for machine learning algorithms and provide results in a better time period using fewer resources, we implemented transfer learning techniques on embedded systems as well. We utilize the system anxious with the data put away locally on the device and register the handling time. We use Ubuntu as our application with four primary cycles. We utilized the model multiple times and determined the normal time that included model handling, preparing, and testing. The normal time determined for the coronary failure forecast is around 40 seconds, and the mushroom request expectation is 48 seconds.

In addition, we utilize an embedded system for TL technique. We use this two-processor controller by imitating the transfer learning model used on a third-party device and connecting it to a stand-alone program to request data from various locations and obtain a request to initiate the process. What's more, to send back the yield results as a reaction to the edge from the installed framework. We glance back at both of these projects. subsequently, we require the assessment of time taken for the association among two machines by handling the installed arrangement of results. The total analysis time for heart attack prediction is approximately 37 seconds, and mushroom demand prediction is 40 seconds.

Case Study	Edge Device	Edge Networks
Heart Attack Detection System	40 seconds	37 seconds
Mushroom Demand Prediction	48.30 seconds	40 seconds

Figure 4.17 depicts the variation of compiling time of two different domain problems, i.e., classification (Heart attack detection system) and Regression (Mushroom demand prediction) working on homogeneous transfer learning approaches.

Diagnosis of heart disease; over edge gadget, we determined the working season of 40.35 seconds.

As indicated by the implanted framework, the handling time was 36.30 seconds. The normal chance to associate with the device from edge to the inserted framework was 0.430 seconds. The association time from implanted framework to gadget edge is 0.475 seconds. The absolute time taken by the edge networks is 37.20 seconds, which is proficient than the time taken by the edge system. Whereas, for mushroom demand prediction;

the execution time over nearby framework was 48.30 seconds. Additionally, execution time was 39.45 seconds for the edge organizations, and the normal correspondence moment from the edge gadget to the installed framework was 0.2 seconds. The systems administration period from implanted framework to edge gadget was 0.3 seconds. The

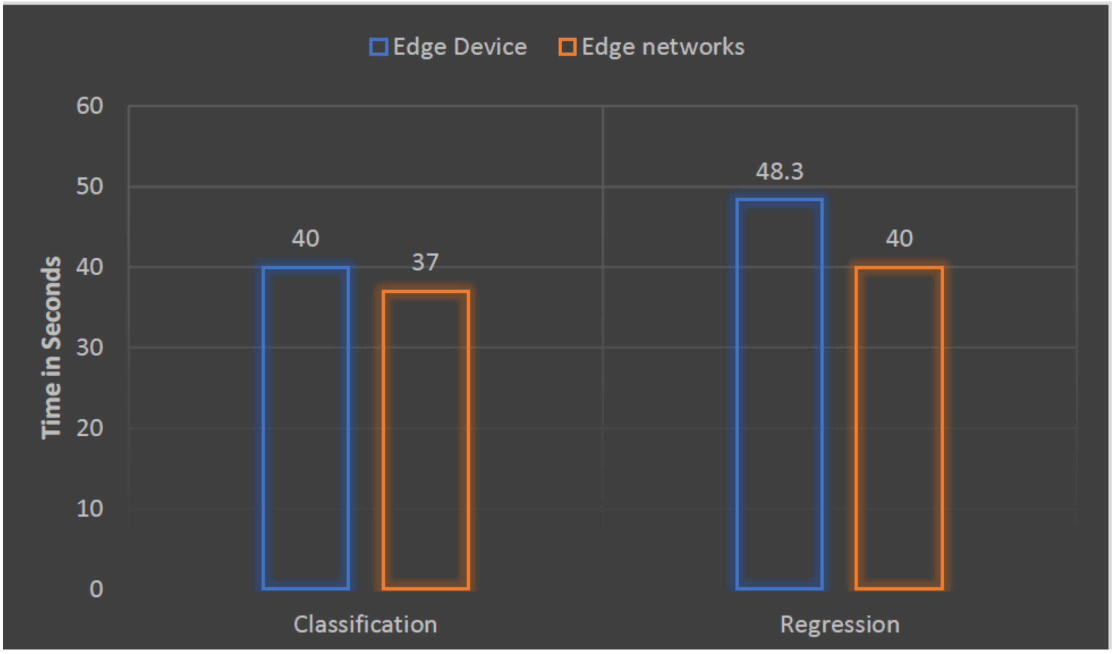

FIGURE 4.17: Computation time of TL over Edge Networks

complete time devoured by the edge gadget was 40 seconds which is outwardly productive than the time taken by the edge gadget. Therefore, using low constraint devices linked with the existing devices in the industry can be considered as a cost-effective and time efficient solution.

Chapter 5

Conclusion

There are so many approaches to move towards new examination in the machine learning region, and notwithstanding, numerous applications are as yet in the dynamic exploration region. Researchers face few challenges while applying machine learning techniques; one of the most significant issues is data security and inaccessible training or testing data, more computation time while compiling and testing, and validation infrastructure requirements. We address the challenge of data availability by implementing various transfer learning approaches. Moreover, we tried to reduce the computation time and lowers the infrastructure requirements by using embedded systems. However, this study implemented different machine learning algorithms to compare the model's outcome accuracy on the classification task. We concluded that the support vector machine and logistic regression work best for the binary classification problem for the heart attack prediction dataset. The least accurate model was the decision tree classifier as per our experiments. Moreover, we implemented the long short term memory model on the periodic data for the mushroom demand prediction domain.

Furthermore, we implemented the homogeneous transfer learning approaches on our two applications, i.e., healthcare safety systems, and the food sales industry. For the classification case study, we implemented feature-based transfer learning. The source domain for this application was breast cancer prediction. The source and target datasets have almost similar parameters and features, and all the data samples have numeric values. Implementing a feature-based approach gave us the accuracy of 94.7% for the target domain as shown in Chapter 4. Furthermore, we first tried implementing the feature-based transfer learning approach for the regression task of mushroom demand prediction. However, the results were not sufficient enough as we got the RMSE score of

0.83; hence, we applied the hybrid-based transfer learning approach, i.e., the combination of feature-based and parameter-based approaches improved the RMSE score to 0.71. Therefore, the feature-based system for the regression problem showed the negative transfer, and it was removed using the hybrid transfer learning approach. We executed the feature, parameter, and mixed transfer learning approaches and compared the results in section Chapter 4. According to the implementations, if the user is experiencing negative transfer while using feature-based or parameter-based TL approaches, then one should compile the hybrid-based approach. The experiments prove that it will provide better pruning of the model resulting inefficient results and predictions. Hence, it will cut off the negative transfer.

This review proposed a typical method to utilize bit-by-bit learning methodologies for intense organizations using edge networks. We have fostered an assortment of ideas that characterize the use of the transfer learning model like boundary estimation, pruning, calibrating to set the connection between various however related areas. Low power and resource constraint devices are proven in Chapter 4 to work for homogeneous transfer learning approaches. Moreover, we used an embedded system with fewer resources than the local or edge system and networked both together to form edge networks. For additional explanation, we attempted to look at the precision and execution time for both the application models in installed frameworks and embedded systems on edge. Likewise, in another space, we don't need to show without any preparation, which saves a ton of architect time, and will expand the productivity of the model. We implemented both case studies of regression and classification over edge networks and edge devices. We computed the time-stamps for the mushroom demand prediction; time taken by the edge device was 48.30 seconds and compiling time over using embedded systems was 40 seconds. On the other hand, for the heart attack detection binary task, the time taken by the edge device was 40 seconds, and the computing time over edge networks was 37 seconds. It is clear from the results that the edge networks are time and resource-efficient in terms of using any ML models.

The list of some future works are as follows:

- Future research can be expected over heterogeneous transfer learning approaches and unsupervised data samples over the edge networks. As transfer learning analyzes unlabelled data points for the target domain, this can also be helpful in semi-supervised learning. One of the works describe the excellent survey regarding the semi-supervised learning process[52]. Researchers can analyze unsupervised and semi-supervised learning to extract the information from the unlabeled dataset of the target domain using transfer learning features.

- Moreover, another way to approach research can be reducing the model's probability to overfit using the regularization method.

- Since it is proven that transfer learning helps to improve the model's tendency to generalize, there have to be reasons for bigger neural network models on how and why they get generalized.

- After generalization, models have to be robust enough; making advancements in adversarial learning, many researchers have come to various approaches on how they can make models robust, but in the case of transfer learning, one should explore for better practices.

- As transfer learning focuses on the only target domain, it differs from multi-task learning; however, future research can incorporate knowledge of some specific tasks that can be applicable over multi-task learning through transfer learning.

- Although optimization of general transfer learning has been done through various methods like the author of [53] tries to optimize the transfer learning parameters using Bayesian optimization; however, researchers can find some specific scheduling techniques for each of the transfer learning approaches.

- As discussed in the study [2], layer partitioning of neural network models over edge networks is beneficial in terms of efficiency. It is also possible that the layer partitioning of the transfer learning model over embedded systems and fog nodes can also provide better outcomes.

Bibliography

[1] Weike Pan, Evan Wei Xiang, Nathan Nan Liu, and Qiang Yang. Transfer learning in collaborative filtering for sparsity reduction. In *Twenty-fourth AAAI conference on artificial intelligence*, 2010.

[2] Darren Saguil and Akramul Azim. A layer-partitioning approach for faster execution of neural network-based embedded applications in edge networks. *IEEE Access*, 8: 59456–59469, 2020.

[3] Axel Seifert and Stephan Rasp. Potential and limitations of machine learning for modeling warm-rain cloud microphysical processes. *Journal of Advances in Modeling Earth Systems*, 12(12):e2020MS002301, 2020.

[4] Shahadat Uddin, Arif Khan, Md Ekramul Hossain, and Mohammad Ali Moni. Comparing different supervised machine learning algorithms for disease prediction. *BMC medical informatics and decision making*, 19(1):1–16, 2019.

[5] Maoying Li, Liu Yang, and Qinghua Hu. A survey on theories and lagorithms about homogeneous transfer learning. *Nanjing Xinxi Gongcheng Daxue Xuebao*, 11 (3):269–277, 2019.

[6] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big data*, 3(1):1–40, 2016.

[7] Lin Gui, Ruifeng Xu, Qin Lu, Jiachen Du, and Yu Zhou. Negative transfer detection in transductive transfer learning. *International Journal of Machine Learning and Cybernetics*, 9(2):185–197, 2018.

[8] Deepak Saggu and Akramul Azim. Transfer learning on the edge networks. In *2021 IEEE International Systems Conference (SysCon)*, pages 1–8. IEEE, 2021.

[9] Grigorios Tsoumakas. A survey of machine learning techniques for food sales prediction. *Artificial Intelligence Review*, 52(1):441–447, 2019.

[10] Qiong Chen, Zimu Zheng, Chuang Hu, Dan Wang, and Fangming Liu. Data-driven task allocation for multi-task transfer learning on the edge. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 1040–1050. IEEE, 2019.

[11] Sunitha Cheriyan, Shaniba Ibrahim, Saju Mohanan, and Susan Treesa. Intelligent sales prediction using machine learning techniques. In *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, pages 53–58. IEEE, 2018.

[12] Indre Žliobaite, Jorn Bakker, and Mykola Pechenizkiy. Towards context aware food sales prediction. In *2009 IEEE International Conference on Data Mining Workshops*, pages 94–99. IEEE, 2009.

[13] Wei Jiang, Gang Feng, Shuang Qin, and Yijing Liu. Multi-agent reinforcement learning based cooperative content caching for mobile edge networks. *IEEE Access*, 7:61856–61867, 2019.

[14] Li Eckart, Sven Eckart, and Margit Enke. A brief comparative study of the potentialities and limitations of machine-learning algorithms and statistical techniques. In *E3S Web of Conferences*, volume 266, page 02001. EDP Sciences, 2021.

[15] Marwa Mamdouh, Mohamed AI Elrukhsi, and Ahmed Khattab. Securing the internet of things and wireless sensor networks via machine learning: A survey. In *2018 International Conference on Computer and Applications (ICCA)*, pages 215–218. IEEE, 2018.

[16] Wan-I Lee, Cheng-Wu Chen, Kung-Hsing Chen, Tsung-Hao Chen, and Chia-Chi Liu. Comparative study on the forecast of fresh food sales using logistic regression, moving average and bpnn methods. *Journal of Marine Science and Technology*, 20 (2):142–152, 2012.

[17] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[18] Raúl Rojas. *Neural networks: a systematic introduction.* Springer Science & Business Media, 2013.

[19] Oscar Day and Taghi M Khoshgoftaar. A survey on heterogeneous transfer learning. *Journal of Big Data*, 4(1):1–42, 2017.

[20] Muhammad Jamal Afridi, Arun Ross, and Erik M Shapiro. On automated source selection for transfer learning in convolutional neural networks. *Pattern recognition*, 73:65–75, 2018.

[21] Xuhong Li, Yves Grandvalet, and Franck Davoine. A baseline regularization scheme for transfer learning with convolutional neural networks. *Pattern Recognition*, 98: 107049, 2020.

[22] Kasthurirangan Gopalakrishnan, Siddhartha K Khaitan, Alok Choudhary, and Ankit Agrawal. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and building materials*, 157:322–330, 2017.

[23] Tingting Hou, Gang Feng, Shuang Qin, and Wei Jiang. Proactive content caching by exploiting transfer learning for mobile edge computing. *International Journal of Communication Systems*, 31(11):e3706, 2018.

[24] Michael T Rosenstein, Zvika Marx, Leslie Pack Kaelbling, and Thomas G Dietterich. To transfer or not to transfer. In *NIPS 2005 workshop on transfer learning*, volume 898, pages 1–4, 2005.

[25] Lubna Badri Mohammed, Alagan Anpalagan, and Muhammad Jaseemuddin. Energy and latency efficient caching in mobile edge networks: Survey, solutions, and challenges. *IEEE Network*, 2021.

[26] Wei Yang Bryan Lim, Nguyen Cong Luong, Dinh Thai Hoang, Yutao Jiao, Ying-Chang Liang, Qiang Yang, Dusit Niyato, and Chunyan Miao. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3):2031–2063, 2020.

[27] Hui Yang, Qiuyan Yao, and Jie Zhang. A distributed federated transfer learning framework for edge optical network. In *Asia Communications and Photonics Conference*, pages S4C–4. Optical Society of America, 2020.

[28] Jiasi Chen and Xukan Ran. Deep learning with edge computing: A review. *Proceedings of the IEEE*, 107(8):1655–1674, 2019.

[29] Min Jiang, Zhongqiang Huang, Liming Qiu, Wenzhen Huang, and Gary G Yen. Transfer learning-based dynamic multiobjective optimization algorithms. *IEEE Transactions on Evolutionary Computation*, 22(4):501–514, 2017.

[30] Angelo Feraudo, Poonam Yadav, Vadim Safronov, Diana Andreea Popescu, Richard Mortier, Shiqiang Wang, Paolo Bellavista, and Jon Crowcroft. Colearn: Enabling federated learning in mud-compliant iot edge networks. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, pages 25–30, 2020.

[31] He Li, Kaoru Ota, and Mianxiong Dong. Learning iot in edge: Deep learning for the internet of things with edge computing. *IEEE network*, 32(1):96–101, 2018.

[32] Liang Xiao, Yuzhen Ding, Donghua Jiang, Jinhao Huang, Dongming Wang, Jie Li, and H Vincent Poor. A reinforcement learning and blockchain-based trust mechanism for edge networks. *IEEE Transactions on Communications*, 68(9):5460–5470, 2020.

[33] Harshit Daga, Patrick K Nicholson, Ada Gavrilovska, and Diego Lugones. Cartel: A system for collaborative transfer learning at the edge. In *Proceedings of the ACM Symposium on Cloud Computing*, pages 25–37, 2019.

[34] Min Jun Park and Mauricio D Sacchi. Automatic velocity analysis using convolutional neural network and transfer learning. *Geophysics*, 85(1):V33–V43, 2020.

[35] Dapeng Wu, Xiaojuan Han, Zhigang Yang, and Ruyan Wang. Exploiting transfer learning for emotion recognition under cloud-edge-client collaborations. *IEEE Journal on Selected Areas in Communications*, 39(2):479–490, 2020.

[36] Shuo Wang, Xing Zhang, Yan Zhang, Lin Wang, Juwo Yang, and Wenbo Wang. A survey on mobile edge networks: Convergence of computing, caching and communications. *Ieee Access*, 5:6757–6779, 2017.

[37] Orpaz Goldstein, Mohammad Kachuee, and Majid Sarrafzadeh. Decentralized knowledge transfer on edge networks for detecting cancer in images. In *2021 IEEE*

EMBS International Conference on Biomedical and Health Informatics (BHI), pages 1–5. IEEE, 2021.

[38] Yunlong Lu, Sabita Maharjan, and Yan Zhang. Adaptive edge association for wireless digital twin networks in 6g. *IEEE Internet of Things Journal*, 2021.

[39] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3712–3722, 2018.

[40] Zitong Wan, Rui Yang, Mengjie Huang, Nianyin Zeng, and Xiaohui Liu. A review on transfer learning in eeg signal analysis. *Neurocomputing*, 421:1–14, 2021.

[41] Aditya Khamparia, Deepak Gupta, Victor Hugo C de Albuquerque, Arun Kumar Sangaiah, and Rutvij H Jhaveri. Internet of health things-driven deep learning system for detection and classification of cervical cells using transfer learning. *The Journal of Supercomputing*, pages 1–19, 2020.

[42] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[43] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4109–4118, 2018.

[44] Juan Zhao, Sachin Shetty, and Jan Wei Pan. Feature-based transfer learning for network security. In *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*, pages 17–22. IEEE, 2017.

[45] Bo Yang, Omobayode Fagbohungbe, Xuelin Cao, Chau Yuen, Lijun Qian, Dusit Niyato, and Yan Zhang. A joint energy and latency framework for transfer learning over 5g industrial edge networks. *IEEE Transactions on Industrial Informatics*, 2021.

[46] Carlos Ordonez. Association rule discovery with the train and test approach for heart disease prediction. *IEEE Transactions on Information Technology in Biomedicine*, 10(2):334–343, 2006.

[47] Hao Xue, Du Q Huynh, and Mark Reynolds. Ss-lstm: A hierarchical lstm model for pedestrian trajectory prediction. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1186–1194. IEEE, 2018.

[48] Eric W Weisstein. Sigmoid function. *https://mathworld. wolfram. com/*, 2002.

[49] Bharath Raj. A simple guide to the versions of the forward propagation. *Retrieved from Towards Data Science website: https://towardsdatascience. com/a-simpleguide-to-the-versions-of-the-inception-network-7fc52b863202*, 2018.

[50] Opetunde Adepoju. A step-by-step tutorial on coding neural network logistic regression model from scratch, Aug 2019. URL `https://medium.com/@opetundeadepoju/a-step-by-step-tutorial-on-coding-neural-network-logistic-regression-model-from-`

[51] Vaishali R Patel and Rupa G Mehta. Impact of outlier removal and normalization approach in modified k-means clustering algorithm. *International Journal of Computer Science Issues (IJCSI)*, 8(5):331, 2011.

[52] Xiaojin Jerry Zhu. Semi-supervised learning literature survey. 2005.

[53] Sebastian Ruder and Barbara Plank. Learning to select data for transfer learning with bayesian optimization. *arXiv preprint arXiv:1707.05246*, 2017.