Real-Time Crop Disease Detection and Remedial Suggestion through Deep Learning-based Smartphone Application

Kavita Pandey\textsuperscript{a} and Dhiraj Pandey\textsuperscript{b,*}

\textsuperscript{a}Jaypee Institute of Information Technology, Noida, India
\textsuperscript{b}JSS Academy of Technical Education, Noida, India

Abstract

More than half of the workforce of many countries, such as India, are still engaged majorly in agriculture, according to a survey. Crop diseases are a major threat to food security that farmers grow every year. The early identification of crop disease remains difficult in many parts of India due to the lack of the necessary infrastructure. Several solutions have been devised at the governmental level to address the challenge of food security. Still, most Indian farmers do not have sufficient technical support to address major problems like monitoring fields, which includes irrigation control, soil moisture, invigilating water level, and detection of crop diseases. A solution in an affordable form that satisfies the Indian context is highly needed. In this article, the issue of crop disease detection has been addressed using the advanced technologies that can be provided in low-cost smartphones. Timely identification of diseases and subsequent immediate remedial action will help in saving the yields which automatically saves the economy of the farmer and in turn can help several farmers from distress. A deep learning-based real-time solution has been proposed that ensures ease of access, convenient architecture, and 24*7 connectivity by empowering the user with the element of Disease Prediction and Remedy suggestion.

Keywords: deep learning; Crop Disease; disease detection; smartphone application; agriculture; prediction system

© 2023 Totem Publisher, Inc. All rights reserved.

1. Introduction

Machine Learning, defined as the capacity of machines to learn, decipher and carry on by understanding the rationale and measurements of the problem, is another recent approach to solving technical issues. Machine Learning is one of the innovations which made human work simpler and provides a superior route for flawlessness. Smart farming fueled by Machine Learning with highly accurate calculations is another idea developing in recent times [1,2].

In today’s time, farmers are facing some of the biggest difficulties in the sector’s history as the global population grows and so does the requirement for more and more food from fewer sections of land. Alongside these difficulties additionally comes the innate challenge for land and water – this issue is likewise being enhanced by labor deficiencies, climate change, and expanding ecological guidelines [3,4]. Diseases that affect crops are a significant danger to food security, yet their fast detection stays troublesome in numerous parts of the world because of the absence of an important foundation. The blend of expanding worldwide cell phone entrance and ongoing advances in computer vision made conceivable by deep learning has led to preparations for cell phone-assisted crop disease detection.

Farming has seen various innovative changes in the most recent decades, getting progressively innovative and industrialized [5,6]. Crop management devices have started to utilize IoT and machine learning in agriculture. Much the same as climate stations, they ought to be set in the field to gather information specific to crop farming. The information related to crops include temperature, relative humidity, soil moisture, pH level, overall crop health, and many more [7]. Along these lines, the crop growth can be monitored and any abnormalities can be avoided to prevent any ailments or invasions that can hurt the yield. Utilizing this, various answers in terms of solutions for crop suggestion and irrigation control have been proposed using crop management devices. These solutions utilize machine learning where machines can work on the collected data and analyze it to reveal hidden useful information beneficial for crops. Machine learning algorithms along with computer vision techniques can be used to identify various issues that arise during crop growth, such as weed and disease detection.

\textsuperscript{*} Corresponding author.
E-mail address: dhp2@yahoo.co.in
Appropriate measures could be suggested in these situations [8]. These technologies can also help us in predicting crop yield in the field of agriculture.

These innovations and proposed solutions are being used in many countries. However, in India, it has been quite difficult to widely employ these technologies. Though India is primarily an agrarian country, the farmers are still not financially equipped to afford such technologies on a wide scale because the majority of Indian farmers own small fields and grow a single crop in one season. Therefore, a solution is required in India which is beneficial for small farmers and provides them with a small and affordable form factor. This article tries to address all these needs in our proposed solution.

This article proposes a deep learning-based system that collects real-time plant images from the user’s mobile application and analyzes those images for disease prediction and suggests appropriate remedies. The primary objective of this work is to provide an easy and quickly accessible platform by which plant health can be easily monitored by the farmers. The proposed solution uses a deep learning model to identify and categorize plant diseases. The platform also suggests the appropriate remedies as per the identified disease. Flutter has been used to deploy an Android application. It makes use of proposed deep learning models which provide an easy to use interface as well as offline functionality. Section 2 discusses the research on this problem. The proposed work along with its architecture has been elaborated on in Section 3. The underlying technologies to build the proposed system and other implementation details have been presented in Section 4. It is followed by Section 5, which depicts the evaluation of the system’s performance through appropriate results. Finally, concluding remarks and future possibilities have been discussed in Section 6.

2. Related Work

To handle the issues mentioned in the above section, some research has been done in a similar direction to provide meaningful agricultural solutions. This section discusses the relevant prior research focusing on agricultural data analysis based on machine learning and makes use of that data for various purposes, such as disease prediction or various other predictions. Some researchers have utilized the deep learning model to detect a specific type of disease, while others have focused on creating a dataset for further development.

Mohanty et al. [9] have used a public dataset containing almost fifty thousand images of diseased and healthy plants. The authors used the deep convolutional network for training and prediction. They can recognize 26 diseases spanning over 14 crops with a high accuracy of 99.35%. The authors also experimented with various techniques involving segmentation ratio, and colored or non-colored images to determine the best type of model that can be used for the plant disease diagnosis system. In another work done by Vanitha et al. [10], the authors proposed automatic plant disease detection by using a deep convolutional network. The utilized dataset contains 500 images of rice crops, including both healthy and diseased crops. The model was used to train three common diseases of the rice crop and the authors achieved an accuracy of 99.53%.

Davinder et al. [11] have explored the use of computer vision for disease detection in plants. The main aim of this study was to develop a large-scale non-lab dataset for deep learning usage. They developed a dataset containing more than 2500 data points and up to 17 classes of diseases by annotating internet-scraped images. The results denoted that the developed dataset, when modeled, can increase the classification accuracy by up to 31%. In a similar direction, Arsenovic et al. [12] have proposed a dataset containing roughly 80,000 images with the goal being to turn it into the biggest dataset containing leaf pictures. Pictures were taken in different climate conditions, at various angles, and at different times of the day with backgrounds that looked closer to real-life scenarios. The two significant methodologies that were utilized to expand the dataset were traditional augmentation methods and state-of-the-art-style generative adversarial networks. A few experiments were done to test the effect of training, in a controlled domain and utilization in real-life situations to precisely recognize plant diseases in an unpredictable background and different conditions including the identification of numerous sicknesses for a single leaf, and the trained model was found to have an accuracy of 93.67%. The referred literature has been compared and presented in Table 1 for the reader’s reference.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dataset used</strong></td>
<td>Public</td>
<td>Developed</td>
<td>Public</td>
<td>Public + Developed</td>
</tr>
<tr>
<td><strong>(Public/Developed)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Number of images</strong></td>
<td>50000 containing 14 crops and 26 related diseases.</td>
<td>2500 containing 17 crop diseases</td>
<td>500 images containing 3 diseases related to the rice crop.</td>
<td>80000 images aimed to be the largest dataset.</td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td>99.35%</td>
<td>No specific study was done.</td>
<td>99.53%</td>
<td>93.67%</td>
</tr>
<tr>
<td><strong>Mobile Application</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Real-Time Crop Disease Detection and Remedial Suggestion through Deep Learning-based Smartphone Application

Our proposed work addresses major drawbacks in the related literature by employing all the parameters that were missing in the previously mentioned articles. The proposed work considers a large dataset of plant leaves which includes both diseased and healthy leaves. It helps us in creating a highly accurate model. The work is integrated through a mobile application, which provides ease of access to farmers. The following section discusses the architecture and other details of the proposed work.

3. Proposed Solution

The proposed solution is a simple and easily deployable system that predicts the diseases of crops and provides remedial solutions. It does the disease prediction through an in-app functionality based on a trained model using a deep convolutional network. Remedial solutions would be suggested for any disease if detected by the system. Users can provide any real image and the system detects the disease, if any. To make the system user-friendly, a mobile application has been provided alongside the website interface so that farmers can easily interact with and control the system. The complete architecture of the proposed work has been described in Figure 1 which is further illustrated in the next section.

![Figure 1. The modular architecture of the proposed system](image)

4. Implementation Details

A deep convolutional neural network to recognize 9 crop species and 33 diseases (or nonappearance thereof) has been trained. The convolutional neural network (CNN) is a class of deep learning neural systems that are used to perform the task. They are most usually used to examine visual imagery and are most of the time used in image classification working behind the scenes. The internal architecture of CNN is described in Figure 2.

![Figure 2. Internal Architecture of Convolutional Neural Network](image)
The workings of CNN [14] are described as follows through the following steps:

- An input image is first provided to the convolution layer.
- Choose the appropriate parameters. If required, apply the filters with strides with padding.
- Image convolution is performed and ReLU activation is applied to the matrix.
- To lessen the dimensionality size, pooling would be performed.
- As per the requirement, several convolutional layers can be added.
- The output is then flattened and fed into a fully connected layer (FC Layer)
- Then the class is outputted using an activation function (Logistic Regression with cost functions) and categorizes the pictures.

The pictures of plant leaves are examined, which have a spread of 33 class labels designated to them. Each class label is a crop-disease pair, and an effort is made to predict the crop-disease pair by giving only the picture of the crop leaf. It can be visualized from Figure 1 that the proposed system utilizes two datasets. Dataset 1 has been used for training the model. Once the model gets trained, it utilizes Dataset 2 for providing the remedial solution for the user-inputted image.

4.1 Datasets Used

**Dataset 1:** To create accurate image classifiers for plant infection diagnosis, an enormous and verified dataset of images of infected and healthy plants was required. Only a few datasets exist and most were not openly accessible. To address this issue, Ali [15] made a Plant Village project. The objective of this project was to gather a huge number of images of healthy and infected crop plants and to make them directly and openly accessible. Here, we report on the characterization of 33 diseases in 9 crop species utilizing more than 10,000 pictures with a convolutional neural network approach. The nine crops are Apple, Cherry, Corn, Grapes, Pepper, Potato, Strawberry, Tomato, and Peach. Each type of crop has at least some type disease, resulting in a total of 33 diseases. For example, the Apple crop suffers from black rot and apple_rust, whereas corn suffers from Cercospora_leaf_spot and Common rust. Similarly, all other crops were suffering from some amount of diseases. Overall, there are 33 classe s as per the total number of diseases.

**Dataset 2:** A dataset containing the necessary remedies and solutions for crop diseases was implemented to act as a handbook or as an intermediate solution to control the growth of those diseases or to eliminate them. The dataset includes remedies for 33 diseases of 9 crops as mentioned in dataset1 [16][17].

To address the problem of over-fitting, the test set to train set ratio was chosen. The model was trained on 80% of the information and tested the prepared model on the other 20% of the dataset. This was the same as the literature survey, since such conditions accomplished the maximum accuracy. A trained model was then created using the above-mentioned layers of CNN and the model weights were then saved to avoid the model’s re-training. The file was saved in h5 format [18]. This helps to easily implement the saved model in various scenarios, as the model can be easily loaded from the saved weights and could be used further for disease prediction purposes by checking it with a user-supplied image.

This trained model was then made to communicate with our separate app for disease detection by converting it to the TFlite version also known as TensorFlow Lite version as it is made for mobile versions [19]. With this, we can provide the functionality for the user to upload images of the sown crops so that the system could predict if the crop has been affected by any diseases and provide the corresponding solution.

4.2 Application Creation using Flutter

The C and C++ code is compiled with Android's NDK [20]. The Dart code is assembled into local, ARM, and x86 libraries. Those libraries are imported for a "runner" Android project, and the entire thing is incorporated with an APK. An image picker was used to get the clicked photograph from the mobile gallery. When launched, the application stacks the Flutter library [21]. The following section discusses the results of the proposed system.

5. Results

As discussed in the above sections, the proposed solution predicts plant disease as well as suggests a remedial solution. For the first phase, which is the disease prediction part, the model has been trained and tested. Thus, initially, in this section, results related to that part would be discussed. It is followed by the results of the testing of a real inputted image and its solutions.
Figure 3 depicts the training accuracy after each epoch as observed on the terminal screen of the executed code. It can be visualized from Figure 3 that during the testing of the dataset, our accuracy reaches up to 97.85% after the 100th epoch. This accuracy helps to give an estimate of how accurate our trained model is. Figure 4 shows the graphical representation of the accuracy of the model being trained after each epoch. Each epoch refers to one cycle through the full training and is when the whole dataset is passed forward and in reverse through the neural system once.

![Figure 3. Statistics showing the accuracy achieved after each epoch](image1)

![Figure 4. Graph for Training and Validation accuracy and Training and Validation loss](image2)

It can be visualized from Figures 3 and 4 that high accuracy has been achieved by the disease detection model when trained using CNN. To validate the model, different images have been provided and system behavior has been analyzed. For this, the trained model has been imported as shown in Figure 5. The trained model against different images would be checked.

![Figure 5. Loading of trained model for prediction](image3)
Figure 6. Detection of Black Rot disease of Apple crop

Figure 7. Detection of Leaf Mold disease of Tomato crop

Figure 8. Detection of Powdery Mildew disease of the Cherry crop
It can be visualized from Figures 6, 7, and 8 that different crop images have been provided as input for testing the system. The system can predict the correct diseases with their labels. In Figure 6, an image named “Apple__Black_rot_021.jpg” is provided to the system. This image is of an Apple crop infected with Black rot disease and we want our system to detect the same. Thus, after resizing the image, it is provided for the next step. The trained model checks this image and provides the output as class label 2 and named Apple__Black_Rot. It can be visualized in the output terminal of Figure 6; thus, this output verifies the accuracy of our model.

Similarly, another image named “Tomato__Leaf_Mold_012.jpg” is provided to the system for verification. This image is from the Tomato-Leaf Mold crop-disease pair. The system checks this image and provides the output with the class label 28 and the name Tomato Leaf Mold. The model detects that this leaf is from a tomato crop and suffered from Leaf Mold disease. The same has been displayed in the output terminal, which again verifies the accuracy of our model. As another test case, an image named “Cherry__Powdery_mildew002.jpg” is provided to the system. This image is from the crop-disease pair Cherry-Powdery Mildew. Our model also detects that this image suffered from Powdery Mildew disease, and it is from a Cherry crop. It is generating the output as class label 6 and name Cherry Powdery as displayed in the output terminal of Figure 8. This verifies our model works and predicts the correct crop-disease pair when provided with an input image with more than 98% accuracy that is also verified by providing various single images as input.

Testing of mobile applications for disease detection and suggestion has also been done. This can be visualized in Figures 9 and 10. Figures 9 and 10 show the detection of Scab disease in Apple crop and Spider Mites disease in Tomato crop respectively.

In Figure 9, it can be visualized that when an image of an Apple plant affected by Scab disease, it is provided in the application. At the backend, the trained model starts working and the output Apple Scab, along with the corresponding remedy, is displayed on the user screen. Figure 10 also displays a similar kind of output for the Tomato plant, which is affected by Spider Mites disease. It also suggests the corresponding remedial solution for the same.

6. Conclusion and Future Work

Many people are still engaged in agriculture because of factors like lack of education and awareness which does not allow them to shift to the industrial and service sector and social factors like believing that being born in a family of farmers means becoming a farmer. Due to poverty in the farmer community, especially in India, many cannot take advantage of advanced technologies to enhance their yield production. This work aimed to provide a single comprehensive and affordable solution in terms of a mobile application employing deep learning using low-cost smartphones. This would help the farmers in monitoring their fields. During monitoring, they can just click and provide an image of the proposed application; it will detect the disease as well as suggest the remedy. The application has been evaluated on different test cases and the obtained results prove its efficiency.

The application can be enhanced with more features in future work. The application has been tested in ideal conditions, but sometimes the conditions in the field are not ideal and the collected pictures are not very clear. In these kinds of situations,
the model’s precision might be diminished greatly. So, the model should be trained and tested thoroughly in different situations. This model can be trained for more crop datasets in the future. The methodology introduced here mustn’t be proposed to swap existing answers for disease finding, but instead to enhance them. Finally, the proposed approach suggests scenarios where the introduced approach should prove adequate.

References


UWGAN-EnhaNet: Conditional Generative Adversarial Network Inspired Network for Enhancing the Quality of Underwater Images

M. J. Delsey and J. V. Bibal Benifa*

Indian Institute of Information Technology, Kottayam, India

Abstract

Researchers have been focusing on uncovering underwater treasures by overcoming the obstacles of poor quality underwater images. The onset of deep learning methods and the various acquisitions of underwater images paved the way for a lot of explorations. In this paper, architecture based on GAN is proposed to enhance the characteristics of the underwater image by preserving the structure and content of the image. Experiments are executed by utilizing the publicly available UFO-120 and UIEB datasets which include both real undersea images and their corresponding reference images. To boost the performance of the architecture, $L_1$ and content-based loss are combined with $L_{GAN}$. The final enhanced image provides an appealing result in qualitative evaluation whereas the results obtained from PSNR, SSIM, and UIQM metrics demonstrate that the suggested strategy produces improved results when compared with the most recent techniques.

Keywords: generative adversarial network; generator; discriminator; loss function

© 2023 Totem Publisher, Inc. All rights reserved.

1. Introduction

The underwater environment is an ecosystem consisting of massive groups of flora and fauna among which many of them are undiscovered. Many underwater projects have been conducted to explore the underwater environment, which includes marine biology, oceanography, geology, and archaeology. The underwater imagery is captured by various methodologies like scuba diving, snorkelling, underwater housing, underwater drones, and specialized underwater cameras. The hindrance in reaching the deep world underneath the ocean has been overcome by the advancement in ROVs. The obtained imagery is used for analysis, such as classifying and detecting underwater objects. Regardless of the advancement in the acquisition of the imagery, many barriers stand as interruptions in analysing the obtained images. The attenuation of light's effects, scattering, and absorption that alter the color, contrast, sharpness, and clarity of underwater images are the main impediments to processing underwater images. The major complication faced in computer vision is the enhancement of the obtained underwater images where the visual quality hinders object detection, recognition, and classification applications.

Deep learning comes up with an encouraging remedy for the problems faced while enhancing the underwater images. Varied deep learning techniques based on encoder-decoder architecture, Convolutional Neural Networks (CNNs) and Generative Adversarial Network (GAN) based networks are utilized in improving the features of the underwater images. Among the various techniques, GAN has shown satisfying results in substituting the style and translation of images from one style to another [1]. The proposed architecture is the modified version of FunIE-GAN [2] which is based on the conditional GAN.

2. Related Work

Image enhancement is a vital area to be focussed upon as it forms the foundation for further investigation of the image. Even though traditional image enhancement techniques are available for the task, the outset of deep learning methodology has shown remarkable improvement in the field of image enhancement. In this section, two major revolutionary techniques are discussed: CNN-based enhancement methods and GAN-based enhancement methods.

* Corresponding author.
E-mail address: benifa@iiitkottayam.ac.in
To amplify the quality of the underwater image, Sun et al. presented an encoder-decoder-based network design [3]. The process is carried out based on pixel-to-pixel. The distortions in the underwater images are removed by the encoder block implemented using convolutional layers whereas the decoder block retrieves the neglected fine details in the image. Wang et al. developed a CNN-based architecture UIE-Net for improving the quality of underwater images [4]. The architecture resolves color and haze-related distortions from the image. Combination of a pair of neural structures, one assessing the scene depth, the other the background light, is employed by Cao et al. [5]. The network outputs the depth map from which the transmission map is obtained. For further refining the quality, a guided filter is adopted in the last stage. Anwar et al. designed underwater water CNN (UWCNN) to overcome the depressed contrast and color misinterpretation [6,7]. For the purpose of training ten kinds of underwater photos, the author created ten UWCNN models. To remove the distortion, Li et al. devised a Water-Net and also constructed a dataset containing real underwater images and their corresponding reference image [8]. Water-Net is a multi-scale gated fusion-based CNN [8]. This model learns three confidence maps that are utilized in enhancing the image.

Instead of collecting classical training data, Li et al. introduced a new approach to creating underwater images from in-air and their depth map by feeding them to WaterGAN [9]. The author also proposed a network that aids in correcting the color of the real underwater image which generates the enhanced image equivalent to an image taken in air. Similar to [9], Fabbrì et al. employ CycleGAN [10] to create a paired set of images for training the model [11]. The proposed UGAN network uses an encoder-decoder design in which U-Net is used in the encoder. Wasserstein GAN is utilized in the discriminator. Following the GAN model, UWGAN was designed by Li et al. [12]. It used an irresolutely managed method in order to transfer the color content to determine the legitimate enhanced image. In the generated image they tried to retain the structure and content as in the input image except for the color. To achieve the expected output, multi-term loss consisting of SSIM loss, cycle consistency loss, and adversarial loss were applied. Multiscale Dense-GAN was proposed by Guo et al. [13] in which a union of residual learning, concatenation in dense format, and multiscale were employed to help in fixing the color cast problem. Hanyu Li et al. introduced a fusion-based new approach to the generative adversarial network (FGAN) which includes various inputs [14]. In the proposed work they used spectral normalization to strengthen the image quality and also applied a well-developed objective function to preserve the image content. Bringing together the efficiency of dark channel prior (DCP) and CycleGAN Lu et al. created an adaptive underwater image restoration (MCycle GAN) method [15]. To further increase the execution of image restoration, the author incorporated a multi-scale structural similarity index measure [15]. To overcome the challenge of eradicating haze and color cast, Ye et al. developed an architecture based on stacking CGAN. The architecture consists of two GANs one used to correct the hazing effect and the other one to correct the color [16].

3. Methodology

This section explicates the structure of the Underwater GAN Enhancement Network (UWGAN-EnhanaNet) which employs the modified architecture of FUnIE-GAN [2] that is built based on the conditional GAN model. This model utilizes the paired images comprising of the real distorted underwater and its associated reference image for the whole process. The foundation of this model is to enable the generator to produce an enhanced image that is equivalent to that of ground truth or reference image. The generator excels in this mission by competing with the discriminator that catches sight of the distinction between the reference and the created image. The generator and discriminator modules’ loss functions help to refine the weights during the training, leading to generating good quality enhanced images from the distorted underwater input image. The design of the generator and discriminator are shown in Figures 1 and 2.

3.1 Generator Design

Influenced by the efficiency of FUIE-GAN [2] architecture, modified generator architecture is designed to contribute to upgrading the characteristics of the enhanced image. The input picture of a distorted underwater picture is passed through a sequence of down-sampling and up-sampling segments. With a momentum of 0.75, there are five convolutional layers paired with ReLU activation and batch normalization. The distorted underwater image is passed to the generator block, which is convoluted followed by the ReLU activation. The outcome of the above process is max pooled which is then passed through two blocks containing 2D convolution, ReLU, and batch normalization. This is repeated for one more block structure. The final max pooled layer is fed to down-sampling blocks. The outcome from the final down-sampling is given as input to the up-sampling block. In order to keep high-quality spatial information, up-sampling skip connections are utilized to merge the spatial information from the down-sampling channel with the up-sampling path. But the processes of using skip connections have the chance of propagating poor feature representation from the initial layers. To avoid the inclusion of poor features, average computation is done to retain the feature map. The resultant from the average unit is passed through consecutive Convolution-ReLU-Batch Normalization layers. The final convolution layer generates the 256 x 256 x 3-sized enhanced images.
UWGAN-EnhaNet: Conditional Generative Adversarial Network Inspired Network for Enhancing the Quality of Underwater Images

Figure 1. Generator Architecture of UWGAN-EnhaNet
3.2 Discriminator Design

PatchGAN is utilized for the discriminator that only judges the structure of the images patched at the local level [1]. The information received at the patch levels is utilized for discrimination. This grasps the provincial texture and style, which are rich in high-frequency features. The discriminator architecture takes the distorted underwater image together with the ground truth and a generated image, half of each, for training the discriminator to identify the actual and fabricated images. The paired input images are concatenated and fed to the first level of the convolution-ReLU layer followed by three layers of convolution-ReLU-Batch normalization with various filters of sizes 64, 128, and 256. In all the above layers, a stride of 2 and the same padding are employed. The final convolution layer outputs the result as a 16 x 16 x 1.
3.3 Loss Function

The most essential criterion for the GAN models is the loss function to boost the performance of the models. In the application of enhancing the image quality, the main objective of the loss function is to compute how realistic the created imitation image is from the reference image provided to condition the GAN. The generator in the GAN always attempts to lessen the loss, while the discriminator aims to escalate it. The mapping of input image domain A to enhanced image domain (i.e.) the target domain B is learned by the classic conditional generative adversarial network. The loss function utilized by the classic cGAN [17] is represented in the Equation 1:

\[ L_{cGAN}(G, D) = E_{A\sim\mathcal{D}}[\log D(B)] + E_{A\sim\mathcal{N}}\log(1 - D(A, G(A, R_N))) \]  

(1)

Here, A is the input domain, which is the distorted underwater image, B is the target domain, which implies an improved version of the undersea image, and \( R_N \) is the random noise. Motivated by the model’s success as demonstrated by the FunIE-GAN loss function [2], the same loss function is employed in the proposed GAN model. To improve the capability of generating the enhanced image on par with the reference/ground truth image, \( L_{L1} \) and content-based loss is combined with \( L_{cGAN} \).

The main goal of the content loss is to improve the generator’s capacity to produce an improved image that is equal to the original image. The difference between the ground truth content and the generated image content, expressed as a mean square, is the content loss. The network of the trained VGG-19 is used for gaining focus on high-level features. Equation 2 depicts the \( L_{con,L}(G) \) [18,19] loss function:

\[ L_{con,L}(G) = E_{A\sim\mathcal{L}}[\log(1 - D(A, G(A, R_N)))][B - G(A, R_N)]^2 \]  

(2)

Here, B is the reference image and \( B' \) is the fabricated better image produced by the generator.

To concentrate on the low frequency features, \( L_{L1} \) loss function is applied, which gives the mean of the absolute variation between the generator produced fake image \( B' \) and the reference image B. Equation 3 represents the \( L_{L1}(G) \) [1] loss function:

\[ L_{L1}(G) = E_{A\sim\mathcal{L}}[|B - B'|] \]  

(3)

Where A is the source image, B is the ground truth image, and \( B' \) is the generated image.

Combining the content loss and mean absolute error with the weight of 0.3 and 0.7 respectively in the loss function of the conditional adversarial network increases the performance of the model and the feature of the generated image will be in good relevance to the reference image provided to condition the model [2].

4. Results and Discussion

To investigate the effectiveness of the suggested approach, a quantitative comparison is conducted on modern deep learning techniques. The experiment is done using Keras with TensorFlow 2.9.1 on a processor with a configuration of Intel Core-i7 CPU, graphics card of NVIDIA GeForce RTX 3060 with a capacity of 6GB using CUDA 11.5 and CUDNN 8.3.1. An ADAM optimizer with a learning rate of 0.0003 is used to train the model. The model was trained using a batch size of 4 and 50 iterations.

4.1 Dataset

For determining the accomplishment of the proposed work of enhancing the underwater image, the following datasets UFO-120 [20] and UIEB [8] are considered. The UFO-120 dataset consists of about 120 paired images for testing and 1500 paired photos for training and validation. The Underwater Image Enhancement Benchmark (UIEB) [8] dataset consists of 950 actual underwater pictures, among which 890 have the matching ground truth pictures. The UFO-120 and UIEB datasets include considerable samples of bluish and greenish-toned distorted images that are utilized in training the model to generate enhanced images from the mentioned tones as most of the underwater images are distorted with the tones that are blue and green. Samples of the images contained in the UFO-120 and UIEB dataset is displayed in Figures 3. and 4, respectively.
4.2 Qualitative Evaluation

Several samples from the UIEB dataset are taken into consideration in order to qualitatively assess the performance of the suggested technique. Figure 5 specifies the real underwater image in Section A, the matching ground truth picture in Section B, and the proposed UWGAN-EnhaNet generated enhanced image in Section C. The instance constitutes samples from various greenish and bluish tones. The proposed method has reasonably enhanced the distorted image but still lacks in brightness feature from the reference image. It has overcome color cast and blurriness issues in the distorted image. The texture and fine details of the image are retained in the generated image.

4.3 Quantitative Evaluation

The four metrics applied to investigate the proposed method are Peak Signal-to-Noise Ratio (PSNR) [18,21], Structural Similarity (SSIM) [21], Underwater Image Quality Measure (UIQM) [22] and Mean Squared Error (MSE). The picture quality of the created image B’ in comparison to its reference image B is evaluated using PSNR, which is represented in Equation 4:

\[
PSNR(B', B) = 10 \log_{10} \left( \frac{255^2}{MSE(B', B)} \right)
\]
The SSIM metric, shown in Equation 5, is used to determine how closely the generated picture B' and the corresponding ground truth image B are related. For computing the closeness between the images, SSIM considers the structure, luminance and, contrast features of the image.

\[
SSIM(B', B) = \frac{2\mu_{B'}\mu_{B} + c_1}{\mu_{B}^2 + \mu_{B'}^2 + c_1} \cdot \frac{2\sigma_{B'B} + c_2}{\sigma_{B'}^2 + \sigma_{B}^2 + c_2}
\]

(5)

In Equation 5, \(\mu_{B'}\) and \(\mu_{B}\) represents the pixel mean of the generated and ground truth images, \(\sigma_{B'}^2\) and \(\sigma_{B}^2\) are the variance of \(B', B\); whereas \(\sigma_{B'B}\) specifies the cross-correlation of \(B', B\). The constants \(c_1\) and \(c_2\) is represented in Equation 6:

\[
c_1 = (k_1L)^2, \quad c_2 = (k_2L)^2
\]

(6)

Here, the low denominator division is maintained by using the coefficients \(c_1\) and \(c_2\. L\) is the dynamic range of the pixel value which is 255 and \(k_1\) and \(k_2\) are fixed with default values 0.01 and 0.03 respectively [2]. Table 1 compares the average PSNR and SSIM [20] values of UWGAN-EnhaNet with contemporary techniques considering the UFO-120 dataset. Similarly, regarding the UIEB dataset, Table 2 compares MSE, PSNR, and SSIM [8]. Water-Net shows a slight increase when compared to the proposed method but in PSNR and SSIM measures, the suggested approach provides a considerable good measure compared to the other innovative techniques. The measure in red indicates the highest value and green depicts the second highest.

Table 1. Comparing average PSNR and SSIM values quantitatively using the UFO-120 dataset [20]

<table>
<thead>
<tr>
<th>Model</th>
<th>PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRCNN</td>
<td>24.73</td>
<td>0.72</td>
</tr>
<tr>
<td>SRResNet</td>
<td>25.23</td>
<td>0.74</td>
</tr>
<tr>
<td>SRGAN</td>
<td>26.11</td>
<td>0.75</td>
</tr>
<tr>
<td>RSRGAN</td>
<td>25.25</td>
<td>0.79</td>
</tr>
<tr>
<td>SRDMR</td>
<td>26.23</td>
<td>0.79</td>
</tr>
<tr>
<td>SRDRM-GAN</td>
<td>26.26</td>
<td>0.78</td>
</tr>
<tr>
<td>Deep SESR</td>
<td>28.57</td>
<td>0.85</td>
</tr>
<tr>
<td>FusionGAN</td>
<td>24.07</td>
<td>0.82</td>
</tr>
<tr>
<td>FunIE-GAN</td>
<td>25.15</td>
<td>0.82</td>
</tr>
<tr>
<td>UWGAN-EnhaNet</td>
<td>29.03</td>
<td>0.86</td>
</tr>
</tbody>
</table>

Table 2. Comparing average MSE, PSNR, and SSIM values quantitatively using the UIEB dataset [8]

<table>
<thead>
<tr>
<th>Model</th>
<th>MSE</th>
<th>PSNR (dB)</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fusion-based</td>
<td>1.128</td>
<td>17.6077</td>
<td>0.7721</td>
</tr>
<tr>
<td>Retinex-based</td>
<td>1.2924</td>
<td>17.0168</td>
<td>0.6071</td>
</tr>
<tr>
<td>GDCP</td>
<td>4.016</td>
<td>12.0929</td>
<td>0.5121</td>
</tr>
<tr>
<td>Histogram prior</td>
<td>1.7019</td>
<td>15.8215</td>
<td>0.5396</td>
</tr>
<tr>
<td>Blurriness-based</td>
<td>1.9111</td>
<td>15.318</td>
<td>0.6029</td>
</tr>
<tr>
<td>Water CycleGAN</td>
<td>1.7298</td>
<td>15.7508</td>
<td>0.521</td>
</tr>
<tr>
<td>Dense GAN</td>
<td>1.2152</td>
<td>17.2843</td>
<td>0.4426</td>
</tr>
<tr>
<td>Water-Net</td>
<td><strong>0.7976</strong></td>
<td><strong>19.113</strong></td>
<td>0.7971</td>
</tr>
<tr>
<td>UWGAN-EnhaNet</td>
<td>0.7853</td>
<td><strong>19.118</strong></td>
<td><strong>0.8034</strong></td>
</tr>
</tbody>
</table>

Underwater Image Quality Measure (UIQM) is the metric used to assess the image's quality without consulting a reference image [22]. It measures the features including contrast, colourfulness, and sharpness. The outcome of this measure depicts the relevance of human visual perception. Higher UIQM indicates the generated image is of good quality in the viewpoint of sharpness, contrast, and colour. Table 3 indicates the UIQM measure on the UIEB dataset [8] and the UWGAN-EnhaNet comes with the outstanding result compared to the other methods.

Table 3. Quantitative comparison based on UIQM values on UIEB dataset [8]

<table>
<thead>
<tr>
<th>Model</th>
<th>UIQM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fusion-based</td>
<td>1.5310</td>
</tr>
<tr>
<td>Two-step-based</td>
<td>1.4002</td>
</tr>
<tr>
<td>Retinex-based</td>
<td>1.4338</td>
</tr>
<tr>
<td>UDCP</td>
<td><strong>1.6297</strong></td>
</tr>
<tr>
<td>Regression-based</td>
<td>1.2996</td>
</tr>
<tr>
<td>GDCP</td>
<td>1.4301</td>
</tr>
<tr>
<td>Red Channel</td>
<td>1.2147</td>
</tr>
<tr>
<td>Histogram prior</td>
<td>1.5440</td>
</tr>
<tr>
<td>Blurriness-based</td>
<td>1.3757</td>
</tr>
<tr>
<td>UWGAN-EnhaNet</td>
<td><strong>2.3490</strong></td>
</tr>
</tbody>
</table>
5. Conclusion

A modified version of FUNIE-GAN is the model that is suggested in this paper as a means of improving the underwater image. The alteration is done in the generator architecture where an averaging module is introduced. It reduces the chance of propagating poor features from the skip connections. The proposed method generates a good quality image that overcomes various problems in the distorted image, like color, cast, and blurriness. The result produced from the quantitative evaluation also quantifies that the proposed methods notably generate good-quality images. In the enhanced image, the bluish and greenish effect has been removed. This enhanced image can be used for further processes like accurate classification and detection of various objects. The study is done based on the paired images but it is not trained and tested with unpaired images, which will be taken into account in future work.

References

Hybrid Machine Learning Model for Load Prediction in Cloud Environment

Savita Khurana, Gaurav Sharma*, and Bhawna Sharma

Seth Jai Parkash Mukand Lal Institute of Engineering and Technology, Radaur, India

Abstract

Virtual machine (VM) load prediction is a critical task in cloud computing. Accurate VM load prediction can help to improve resource utilization, reduce costs, and enhance the quality of service. For the Hybrid LSTM and AdaBoost model, a novel approach is proposed for accurate VM load prediction in cloud environments. The proposed model combines the power of LSTM and AdaBoost model, aiming to capture temporal dependencies in the VM load data and enhance prediction accuracy. The proposed model leverages LSTM to learn patterns and dynamics from historical load data, while AdaBoost is used to create an ensemble of weak regressors that collectively make load predictions. The model follows a two-step process: first, LSTM is trained on historical load data to extract informative features, and then AdaBoost is trained to combine the predictions from multiple weak regressors. The hybrid model demonstrates improved performance in VM load prediction by effectively handling non-linear relationships, temporal dependencies, and complex load patterns. The outcome of the proposed model is calculated using metrics, i.e., MAE, MAPE, MSE, RMSE, R², and compared with existing machine learning algorithms i.e., AdaBoost, KNN, SVM and deep learning algorithms i.e. LSTM, RNN. The results clearly show the superiority of the proposed hybrid approach in accurately predicting virtual machine load, enabling efficient resource allocation and management in cloud computing environments.

Keywords: load prediction; machine learning; deep learning; LSTM; Adaboost; cloud

© 2023 Totem Publisher, Inc. All rights reserved.

1. Introduction

Virtual machine (VM) allocation is a crucial task in cloud environments which affect the efficacy and cost-effectiveness of the cloud performance. Service providers aim to enhance resource utilization, while customers seek cost-effective resource allocation. However, varying workloads and diverse quality of service (QoS) requirements result in fluctuating resource demands. Latency in resource availability further complicates things. Efficiently supplying adequate resources with assured QoS is essential for cloud providers and users [1,2]. One approach to address this challenge is accurate prediction of future workload behavior. Analyzing past usage data enables predictive analysis and better decision-making in resource management. The workload prediction is particularly crucial for application execution efficiency in clouds [3-5]. Increased CPU load negatively impacts performance and reliability. Therefore, automated and adaptive approaches are necessary to predict load demand for efficient resource allocation. However, predicting load for multiple VMs in cloud environments has received less attention, despite the interconnected nature of modern cloud-based software [6]. Accurate VM load prediction helps optimize resource utilization, reduce costs, and improve QoS. Various machine learning methods have demonstrated effectiveness in VM load prediction, [7,8]. In this paper, a hybrid machine learning mode has been proposed that combines the strengths of LSTM and AdaBoost for virtual machine load prediction. In comparison to traditional machine learning models [9], the hybrid model combines LSTM and AdaBoost to specifically address the challenges associated with VM load prediction. Traditional machine learning models often struggle to capture the intricate temporal dependencies and patterns in the load data. In contrast, the proposed hybrid model effectively incorporates LSTM to capture and model these dependencies, resulting in more accurate load predictions. The deep learning models, such as LSTM, excel at capturing temporal dependencies, but they encounter difficulties in handling noisy or outlier data points. The hybrid model overcomes this limitation by integrating AdaBoost, which excels at handling noisy data by assigning higher weights to challenging-to-predict samples, leading to more accurate load predictions even in the presence of irregularities in the data. Moreover, the proposed hybrid model provides greater flexibility and interpretability compared to deep learning models. Deep learning models often
have complex architectures with numerous parameters, making it challenging to interpret and comprehend the inner workings of the model. In contrast, the proposed model combines the simplicity and interpretability of AdaBoost with the temporal modeling capabilities of LSTM. This enables a better understanding and interpretability of the model, which is crucial in cloud environments where transparency and understandability are important factors. The hybrid model for VM load prediction effectively combines the strengths of LSTM and AdaBoost to overcome the drawback of existing models.

1.1 Objective

The objectives of the work are shown below:

- Investigating the complementary strengths of LSTM and AdaBoost in capturing both short-term fluctuations and long-term trends in virtual machine load patterns.
- Designing a hybrid model architecture that effectively integrates the outputs of LSTM and AdaBoost to generate accurate load predictions and effectively capture the temporal dependencies and patterns present in VM load data.
- Evaluating the proposed model performance using different relevant metrics with existing algorithms based upon machine learning techniques i.e. AdaBoost, SVM, K-nearest, and algorithms for deep learning i.e., LSTM, to assess the improvement achieved through the combination of these techniques.

1.2 Problem statement

The aim is to present a machine learning model that combines the strengths of LSTM and AdaBoost algorithms for accurate prediction of future VM load. Let \( L = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\} \) be the dataset, where \( x_i \) represents the input features of the VM at time step \( i \), and \( y_i \) represents the corresponding VM load at time step \( i \). The dataset is split into a training set \( L_{\text{train}} \) and a test set \( L_{\text{test}} \). The objective is to train a hybrid LSTM-AdaBoost model that takes the historical input features \( x_i \) as inputs and predicts the future VM load \( y_{i+1} \) at time step \( i+1 \). The model should effectively capture the temporal dependencies and patterns in the VM load data and handle noisy or outlier data points for improved prediction accuracy.

The problem can be defined as follows:

- Training dataset \( L_{\text{train}} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_M, y_M)\} \)
- Test dataset \( L_{\text{test}} = \{(x_{[M+1]}, y_{[M+1]}), (x_{[M+2]}, y_{[M+2]}), \ldots, (x_N, y_N)\} \)

Proposed hybrid model \( F \) that predicts the future VM load \( y_{i+1} \) at time step \( i+1 \), given the input, features \( x_i \). The main objective is to minimize the prediction error between the predicted VM load \( y_{i+1} \) and the actual VM load \( y_{i+1} \) in the test dataset \( L_{\text{test}} \).

2. Related work

A genetic algorithm-based model for estimating CPU and memory use for virtual machines was proposed by Zeng et al. [10] and compared with the gray model, the results of the method show more accurate predictions, especially for stability and instability. Shyam and Manvi [11] presented a forecasting model that addresses short-term and long-term resource usage of CPU/memory in virtual environments. They make use of Bayesian networks to capture the relationships and dependencies between variables, for more accurate predictions of desired outcomes. Lu et al. [12] introduced a model prediction function based on the Back Propagation Neural Network (BPNN) algorithm, named RVLBPNN (Random Variable Learning Rate Backpropagation Neural Network). Models that take into account CPU-intensive and memory-intensive workloads and use intrinsic workload relationships outperform Hidden Markov Models and Naive Bayesian classifiers in predictive accuracy. Zhang et al. [13] proposed a weather forecasting method using a deep belief network (DBN) that improves forecast accuracy with techniques such as orthogonal experimental design and Spot the Difference. A comparison with the ARIMA model shows that the forecast accuracy of CPU and RAM demand forecasting has increased significantly. However, machine learning methods are often complex and require a lot of data for extraction and training models, which can be lengthy and limit their use for quality of service (QoS) assurance.

Calheiros et al. [14] proposed an ARIMA model for predictions of cloud operations, emphasizing the importance of identifying user behavior in an attempt to truly accurately reflect the world. The ARIMA model was used to create a short-term forecasting model for cloud operations, highlighting the significance of understanding user behavior in an effort to actually accurately reflect reality. To increase forecast accuracy while cutting down on calculation time, the suggested method
integrates ARIMA models with EEMD and RT algorithms. The consideration for factors like estimating error, validity, and time cost evaluation, are used to compare the EEMD-ARIMA with ARIMA models. The result shows an improvement in performance prediction, enabling improved resource allocation and QoS optimization in cloud systems. Jiang et al. [15] model combined various prediction methods such as the autoregressive and QoS model, a model based upon the ann and svm model. The weight of each method is dynamically changed according to the relative error to achieve the best average performance. The Empirical Mode Decomposition (EEMD) technique is used to control the non-stationary hosts, thus increasing the accuracy of the prediction while reducing the computation time. To address the problem of estimation in the context of cloud resource allocation, Xu et al. [16] presented fuzzy logic-centered model to predict the load of applications on the web. Rao et al. [17] explores the use of augmented learning with artificial neural networks (ANNs) to optimize the CPU and memory configuration of virtual machines (VMs) to improve application performance. However, these solutions generally focus on the management of CPU resources and do not include other aspects of resource allocation and optimization. Caron et al. [18] proposed a pattern-matching method to identify similar usage patterns in past usage trajectories by cloud users. They interpolate and predict future prices based on similar models. Kundu et al. [19] used the ann model and support vector regression to present a model for capturing the relationship between resource distribution and efficiency in a cloud-virtualized environment. Their model is designed to predict the resources needed to meet performance targets. However, their approach assumes static functionality and stable usage behavior and limits its applicability in the cloud environment. In addition, the offline working model cannot respond to environmental and load changes in a timely manner.

3. Proposed work

A hybrid model of LSTM and AdaBoost based on VM load forecasting is presented and discussed in this section. The proposed hybrid model successfully captures temporal correlations and complicated load patterns in VM load data by fusing the strength of Long Short-Term Memory (LSTM) networks with AdaBoost. The hybrid model provides enhanced accuracy in VM load prediction through a two-step procedure of training LSTM to extract instructive features and training AdaBoost to aggregate predictions from numerous weak regressors. It efficiently manages non-linear interactions and makes use of both LSTM and AdaBoost’s advantages to produce more accurate load estimates. To produce precise and effective predictions, the Hybrid LSTM and AdaBoost model for VM load prediction in the cloud adopts a two-phase methodology.

During the first phase of the method, the proposed model initially makes use of the capabilities of LSTM to identify temporal dependencies and long-term trends in the VM load data. The historical data is gathered and pre-processed, which further feeds into the LSTM model as inputs because the LSTM model is a particular kind of recurrent neural network that excels in learning sequential patterns and memory retention over time. To forecast the VM load for the upcoming time step, it is trained on the pre-processed data. Using the LSTM model, this phase focuses on identifying the underlying dependencies and trends in the VM load data.

In the second phase of the approach, the proposed model integrates the AdaBoost algorithm to further improve the LSTM model’s prediction capabilities. AdaBoost is a boosting method that fuses a number of poor classifiers into one powerful classifier. A distinct subset of the data is used to train each weak classifier. Weak classifiers can be viewed as alternative models in the context of load prediction that produces forecasts based on various characteristics of the data. Based on their performance, the AdaBoost algorithm distributes weights to these weak classifiers, and the final prediction is derived by merging all weak classifiers’ predictions using these weights.

In order to increase the precision of VM load prediction, this phase aims to take advantage of the variety of weak classifiers and successfully combine their predictions. AdaBoost adjusts to the peculiarities of the data and concentrates on the situations where the model is doing poorly by iteratively training weak classifiers and adjusting their weights. Incorporating both strategies, the model seeks to deliver precise and effective forecasts, supporting optimal resource allocation in cloud environments. In Figure 1, the hybrid model’s working architecture is shown below:
3.1 Algorithm

This section presents a proposed algorithm for VM load prediction in the cloud, which combines the power of Hybrid LSTM and AdaBoost techniques. The aim of the algorithm 1 is to provide precise and effective predictions of VM load that will help for planning and efficient allocation of resources in cloud environments. The tabular form Table 1 explains the meaning of all symbols and notations used in Algorithm 1:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>Historical VM load data</td>
</tr>
<tr>
<td>$y$</td>
<td>Corresponding target load values</td>
</tr>
<tr>
<td>sequence_length</td>
<td>Window size for sliding window transformation</td>
</tr>
<tr>
<td>$X_{train}$</td>
<td>Input training data after sliding window transformation</td>
</tr>
<tr>
<td>$y_{train}$</td>
<td>Output training data after sliding window transformation</td>
</tr>
<tr>
<td>$\theta$</td>
<td>LSTM model parameters</td>
</tr>
<tr>
<td>LSTM($\theta$)</td>
<td>LSTM model</td>
</tr>
<tr>
<td>$M$</td>
<td>Number of samples</td>
</tr>
<tr>
<td>$D$</td>
<td>Dimensionality of the VM load data</td>
</tr>
<tr>
<td>$F$</td>
<td>Number of features extracted by LSTM</td>
</tr>
<tr>
<td>$F(X)$</td>
<td>Features extracted from $X$ using LSTM model</td>
</tr>
<tr>
<td>$w_i$</td>
<td>Sample weights for AdaBoost training</td>
</tr>
<tr>
<td>$h_k(F(X_i))$</td>
<td>Prediction of weak regressor $k$</td>
</tr>
<tr>
<td>$\epsilon_k$</td>
<td>Weighted error of weak regressor $k$</td>
</tr>
<tr>
<td>$\alpha_k$</td>
<td>Regressor weight of weak regressor $k$</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>Small constant for stability</td>
</tr>
<tr>
<td>Load_Prediction</td>
<td>Final load prediction</td>
</tr>
</tbody>
</table>

**Algorithm 1**: Hybrid LSTM and AdaBoost Model for VM Load Prediction in the Cloud (HLA)

Input: VM load data: $X \in \mathbb{R}^{N \times D}$, target load values: $y \in \mathbb{R}^{N}$

Output: Load_Prediction $\in \mathbb{R}^{N}$

1. Set $\text{sequence\_length}=5$
2. Create input-output pairs:
   
   $X_{train} \in \mathbb{R}^{M \times \text{sequence\_length} \times D}$, $y_{train} \in \mathbb{R}^{M}$, where $M$ is the number of samples
3. LSTM Model Training:
   3.1 Initialize LSTM parameters: $\theta$
   3.2 Train LSTM model using $X_{train}$ and $y_{train}$ where
   
   $\text{LSTM($\theta$)} = \arg\min_{\theta} \frac{1}{M} \sum_i (y_i - \text{LSTM($X$; $\theta$)})^2$, $\forall i \in \{1, 2, ..., M\}$
4. Feature Extraction:
   4.1 Extract features from $X$ using trained LSTM model where
   
   $F(X) \in \mathbb{R}^{N \times F}$, where $F$ is the number of features
5. AdaBoost Training:
   5.1 Initialize sample weights: $w_i = 1/N$, $\forall i \in \{1, 2, ..., N\}$
   5.2 For each weak regressor $k$:
      5.2.1 Train weak regressor using $F(X)$ and $y_{train}$:
            $h_k(F(X)) = \text{WeakRegressor}_k(F(X))$
      5.2.2 Calculate weighted error:
            $\epsilon_k = \frac{\sum_i (w_i \cdot |y_i - h_k(F(X_i)))|}{\sum w_i}$, $\forall i \in \{1, 2, ..., N\}$
      5.2.3 Calculate regressor weight:
            $\alpha_k = \frac{1}{2} \ln \left( \frac{1 - \epsilon_k}{\epsilon_k + \epsilon} \right)$, where $\epsilon$ is a small constant
      5.2.4 Update sample weights where
            $w_i = w_i \cdot \exp(-\alpha_k \cdot y_i \cdot h_k(F(X_i)))$, $\forall i \in \{1, 2, ..., N\}$
6. Final Prediction:
   6.1 Initialize load prediction: Load_Prediction = 0
   6.2 For each weak regressor $k$:
6.2.1 Predict load values using weak regressor:
    weak_pred = h_k(F(X))

6.2.2 Update load prediction:
    Load_Prediction += α_k * weak_pred

The procedure starts off by setting the sequence length to a fixed value in the LSTM Model Training phase. The data from the VM load is then divided into input sequences that have the same length as the sequences. For each input sequence, corresponding target load values are established. The X_train and Y_train are used to store these input-outputs. Using X_train and Y_train, the method continues to train the LSTM model. The goal is to reduce the $R^2$ error between the target and the predicted value made by the LSTM model. The LSTM model gains the ability to recognize long-term trends and temporal dependencies in the VM load data through training.

After the LSTM Model Training phase, the algorithm moves on to the AdaBoost Training phase. Sample weights are initialized, assigning equal weight to each sample. For each weak regressor, the algorithm trains a model using the extracted features from the VM load data obtained through the trained LSTM model. The algorithm determines the weighted error, which assesses the reliability of the predictions made by the weak regressor. The approach estimates the regressor weight based on the weighted error, giving weaker regressors with higher accuracy a higher weight. The weak regressor’s poor performance is highlighted more in the updated sample weights. In the Final Prediction phase, the load prediction is initialized to zero. The algorithm iterates through each weak regressor, using them to make predictions. The final load prediction is generated by combining the forecasts and weighting them according to their individual regressor weights. The technique takes advantage of the power of numerous weak regressors’ predictions by combining them to produce precise and effective predictions of VM load.

3.2 Complexity computation

The time and space complexity of the proposed Hybrid LSTM and AdaBoost Model for VM Load Prediction in the Cloud algorithm is represented as follows:

3.2.1 Time Complexity

The time complexity of the LSTM Model Training phase is $O(M \times T \times H^2)$, where $M$ belongs to samples, $T$ is related to the length of the sequence, and $H$ is related to hidden units in the LSTM model. During training, the LSTM layers undergo forward and backward passes, which adds complexity. The time complexity of the feature extraction phase, where $N$ is related to samples and $F$ is associated with features, is $O(N \times F)$. This complexity results from the calculations needed to separate the features from the VM load data. The time complexity of the AdaBoost Training phase is $O(K \times N \times F)$, where $K$ is the quantity of weak regressors, $N$ is the quantity of data, and $F$ is the quantity of features. This complexity results from updating the sample weights and training each weak regressor on the retrieved features. The time complexity of the Final Prediction phase, where $K$ is the number of weak regressors, is $O(K \times N)$.

3.2.2 Space Complexity

The space complexity of the LSTM Model Training phase, where $H$ is related to hidden units in the LSTM model, is $O(H^2)$. The LSTM model’s parameters, such as its weights and biases, are stored, which adds to its complexity. The space complexity of the feature extraction phase, where $N$ belongs to samples and $F$ is associated with features, is $O(N \times F)$. The complexity is brought on by having to store the extracted features. The number of weak regressors ($K$) and the size of the training data ($M$) determine the space complexity of the AdaBoost Training phase, which is $O(K \times M)$. This complexity results from the storage of the training and parameter data for each weak regressor. Space complexity for the Final Prediction phase is $O(N)$.

3.3 Preliminaries

A brief description of the models used to compare the proposed work is stated in Table 2 below:
### 4. Results and Discussion

The simulation tests were carried out on a server with an Intel® i7 processor that has four cores and 2.9GHz of clock speed. The implementation was performed in Python version 3.7. Certain libraries and frameworks, such as Tensor Flow, were required for implementing the LSTM model, and libraries for AdaBoost implementation included sklearn ensemble and AdaBoostRegressor. The parameters required for the execution of the proposed work are described in Table 3. Using data from Kaggle, multiple machine learning-based prediction models are analyzed and compared for performance.

#### Table 3. Parameters used for experimental with their values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input nodes</td>
<td>100</td>
</tr>
<tr>
<td>Epochs</td>
<td>20-50</td>
</tr>
<tr>
<td>Batch size</td>
<td>16</td>
</tr>
<tr>
<td>Activation function</td>
<td>ReLU</td>
</tr>
<tr>
<td>Training data</td>
<td>70%</td>
</tr>
<tr>
<td>Number of epochs</td>
<td>500-1000</td>
</tr>
<tr>
<td>Batch size</td>
<td>1-4</td>
</tr>
<tr>
<td>Training algorithm</td>
<td>Adam Optimize</td>
</tr>
</tbody>
</table>

Different models are compared as shown in Table 4 for VM load prediction, using metrics, including Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R² Score.

#### Table 4. Comparison of models using metrics

<table>
<thead>
<tr>
<th>Model</th>
<th>MAE</th>
<th>MAPE</th>
<th>MSE</th>
<th>RMSE</th>
<th>R²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaboost</td>
<td>0.044</td>
<td>0.0136</td>
<td>0.003</td>
<td>0.056</td>
<td>0.985</td>
</tr>
<tr>
<td>KNN</td>
<td>0.183</td>
<td>0.053</td>
<td>0.104</td>
<td>0.322</td>
<td>0.656</td>
</tr>
<tr>
<td>LSTM</td>
<td>0.071</td>
<td>0.020</td>
<td>0.013</td>
<td>0.118</td>
<td>0.953</td>
</tr>
<tr>
<td>RNN</td>
<td>0.114</td>
<td>0.032</td>
<td>0.061</td>
<td>0.248</td>
<td>0.794</td>
</tr>
<tr>
<td>SVM</td>
<td>0.252</td>
<td>0.066</td>
<td>0.311</td>
<td>0.557</td>
<td>-0.0346</td>
</tr>
<tr>
<td>Proposed model</td>
<td>0.0365</td>
<td>0.011</td>
<td>0.002</td>
<td>0.048</td>
<td>0.992</td>
</tr>
</tbody>
</table>

The results show that the Adaboost model achieved the lowest MAE of 0.044, indicating a smaller average difference between the predicted and actual values. It also had a low MAPE of 0.0136, indicating a small percentage error in the predictions. Additionally, the Adaboost model had a low MSE of 0.003 and RMSE of 0.056, indicating good accuracy in the forecasting. The score value of R² is 0.985, indicating that the AdaBoost model achieves a 98.5% of fit in the target variable. The KNN model had a higher MAE of 0.183, indicating a larger average difference between the predicted and actual values compared
to the AdaBoost model. The MAPE of 0.053 suggests a slightly higher percentage error. The KNN model had a higher MSE of 0.104 and RMSE of 0.322, indicating less accuracy in the predictions compared to the AdaBoost model. The LSTM model performed well with an MAE of 0.071 and a lower MAPE of 0.020, indicating better accuracy and a smaller percentage error compared to the KNN model. The MSE of 0.013 and RMSE of 0.118 suggest accurate predictions, and the R2 score of 0.953 indicates a good fit of the LSTM model to the data. The RNN model achieved an MAE of 0.114 and a higher MAPE of 0.032 compared to the LSTM model. The MSE of 0.061 and RMSE of 0.248 indicate less accuracy in the predictions. However, the R2 score of 0.794 suggests a moderate fit of the RNN model to the data. The SVM model had the highest MAE of 0.252 and a higher MAPE of 0.066 compared to other models. The MSE of 0.311 and RMSE of 0.557 indicate less accuracy in the predictions. The negative R2 score (-0.0346) demonstrates the poor performance of the SVM model for fitting in the target variable.

Finally, the Hybrid LSTM and AdaBoost model outperformed all other models with the lowest MAE of 0.0365, indicating the smallest average variance among the actual and predicted outcomes. The model also had a low MAPE of 0.011, indicating a small percentage error. The MSE of 0.002 and RMSE of 0.048 suggest high accuracy in the predictions. The high R2 score of 0.992 indicates that the Hybrid LSTM and AdaBoost model explains 99.2% of the variance in the target variable.

Figure 2. Actual vs predicted load using AdaBoost model

Figure 3. Comparison using KNN model

Figure 4. Comparison using LSTM model

Figure 5. Comparison using RNN model

Figure 6. Actual vs predicted load using SVM model

Figure 7. Residual plot of proposed Hybrid model
The actual and predicted values of the AdaBoost, KNN, LSTM, RNN, SVM, and hybrid model are depicted in Figure 2 to Figure 10, including the proposed model residual plot, true vs predicted load, and a bar chart of target load. The Hybrid LSTM and AdaBoost model showed superior performance in terms of lower errors, higher accuracy, and a better fit to the data compared to the other models. It offers a promising approach for VM load prediction.

5. Conclusion

To enhance the precision and effectiveness of VM load prediction, a hybrid model for VM load prediction is presented. The hybrid model takes the benefits of LSTM and AdaBoost techniques as the long-term dependencies can be learned by the integration of LSTM, and AdaBoost combines the predictions of various weak regressors to get a more reliable and precise final prediction. The model is constructed in two steps: first, an LSTM is trained on historical load data to extract informative features; next, an AdaBoost is trained to combine predictions from various weak regressors, where each regressor focuses on a different component of the load data. The hybrid model successfully manages non-linear relationships, temporal dependencies, and complicated load patterns, demonstrating better efficiency in VM load prediction. The efficiency of the presented model is evaluated using popular metrics and compared the performance with existing machine learning algorithms, such as AdaBoost, support vector machine, K-nearest neighbors, and deep learning algorithms such as LSTM, and RNN. The results demonstrated that the hybrid approach outperforms these algorithms in accurately predicting VM load. Further research is to develop mechanisms for real-time monitoring to maintain the model’s reliability and accuracy over time in dynamic cloud environments.

References


Savita Khurana is working as Assistant Professor at JMIT Radaur Engineering College affiliated to Kurukshetra University, India. She has 15 years of teaching and academic experience and received her Ph.D. degree from Punjab technical university, in 2021. She has published approximately 20 research papers/chapters in various SCI / ESCI/SCOPUS/WoS Journals, Book Chapters, and Conferences. She reviewed various research papers of reputed publication houses such as Springer/Elsevier/IEEE etc. Her research interests include cloud computing, WSN, and Machine learning. savu.khurana30@gmail.com

Gaurav Sharma is working as a Professor and Head of the CSE department at JMIT Radaur Engineering College, affiliated to Kurukshetra University, India. He received his Ph.D. degree from Punjabi University Patiala, in 2015. He has 18 years of teaching and academic experience. He has published approximately 44 research papers/chapters in various SCI/ESCI/SCOPUS/WoS Journals, Book Chapters, and Conferences. He reviewed various research papers of reputed publication houses such as Springer/Elsevier/IEEE etc. His research interests include cloud computing, Fog computing, WSN, and Deep learning. gaurav13@gmail.com

Bhawna Sharma completed her Ph.D. from Deenbandhu Chhotu Ram University of Science & Technology, Murthal Sonepat in 2021. She is working as Assistant Professor at JMIT College, Radaur. She has published approximately 10 research papers/chapters in various SCI / ESCI/SCOPUS/WoS Journals, Book Chapters, and Conferences. Her research interests include cyber security and Machine Learning. bhawnash024@gmail.com
Influence of Lean and Lean Six Sigma on Social Factors in the Moroccan Industry - Case Study

Abdelouahed Boutayeb\textsuperscript{a,*}, Abderrahim Chamat\textsuperscript{a}, Abdelali Ennadi\textsuperscript{a}, and Abdelmajid Daya\textsuperscript{a,b}

\*Industrial Techniques Laboratory, University Sidi Mohamed Ben Abdellah, Fez, Morocco
\textsuperscript{b}Department of Physics, Moulay Ismail University of Meknes, Meknes, Morocco

Abstract

After World War II, the market experienced significant growth, leading to a shift in demand compared to supply. Moreover, the increase in purchasing power resulted in a massive surge in consumption. In order to meet these demands, companies need to satisfy customers in terms of diversity, quality, price, and delivery time. To achieve this, they employ continuous improvement tools such as Lean and Lean Six Sigma (LSS). The primary goal of continuous improvement is to enhance performance in terms of quality, costs, and delivery time. However, these approaches often have negative consequences on social factors such as safety and ergonomics (SE), since they primarily aim to reduce waste and improve economic efficiency. In this article, we investigate the impact of Lean and LSS on SE factors using a case study. We also introduce our new concept of continuous improvement that considers SE social factors. This concept aims to enhance economic performance while ensuring the safety and well-being of workers.

Keywords: Lean; Six Sigma; Lean Six Sigma; economic factors; social factors

© 2023 Totem Publisher, Inc. All rights reserved.

1. Introduction

The industry has long sought to maximize productivity while minimizing costs and ensuring the safety and ergonomics of its workers. To meet these needs, various quality management methodologies have been developed, including Lean and LSS. These methods have been widely adopted in the industry, but their impact on safety and ergonomics factors remains understudied. Industrial companies must demonstrate dynamism by going beyond their core function of production and ensuring they transform into powerful entities capable of ensuring their sustainability. Consequently, to maintain their position in a booming market, improving performance is a continuous challenge that can only be achieved through the application of an approach that evaluates the economic and social performance of the organization [1,2].

Indeed, the effectiveness of the Lean and LSS approach in the industrial domain allows companies to enhance industrial performance, quality, and operational efficiency [3]. Developing countries can employ these techniques in their growth programs to help build a technological foundation for successful industrial development [4]. Morocco is one of the nations that has adopted Lean as a fundamental technique to achieve effective convergence and industrial growth.

By exploring this data, this study aims to provide valuable insights for companies considering adopting Lean or LSS. Additionally, this study could contribute to enriching the literature on the effects of these methods on quality factors in general and offers avenues for future research in this field. The main objective of this study is to analyze the effects of Lean and LSS on safety and ergonomics factors through a case study and descriptive survey. Specifically, we will examine how the implementation of these methods has impacted safety and ergonomics in a particular company. To achieve this, we will adopt a qualitative approach based on interviews and on-site observations. We will also present the principles of our continuous improvement approach called Lean Management Plus (LMP). The LMP approach has been designed to improve production management, optimize safety risks, and address ergonomic constraints.

* Corresponding author.
E-mail address: abdabout81@gmail.com
The first section of the essay provides a summary of the literature review on these two methodologies, Lean and LSS. The second section is to present research and analysis of obstacles to the success of a continuous improvement project, as well as Morocco's approach to overcoming them. Next, we provide a description of our new concept, Lean Management Plus (LMP), and outline the steps for calculating the new performance indicators that we will refer to as the Overall Equipment Effectiveness Plus (OEEP). Finally, we discuss the impact of Lean and LSS on issues such as personnel safety and ergonomics and present a case study of a Moroccan automotive company.

2. Literature Review

The global market has experienced a number of economic and political crises, all of which have been addressed using a strategy of continuous improvement. Continuous improvement is an approach to gradually and continuously enhance a condition to make it more effective [5]. It is a strategy for achieving quality and operational excellence that relies on a process of performance evolution [6]. This methodology includes Lean, which is a systematic approach to eliminating all causes of inefficiencies in value chains. LSS also aims to reduce the causes of errors and defects in business operations [7].

2.1 Historical Background

The history of continuous improvement dates back several centuries and has undergone significant developments over time. The concept of Takt Time was invented in 1930 by the German aerospace industry to better synchronize production with customer demand. Sakichi Toyoda, the founder of the Toyota automotive company, introduced the concept of "just-in-time" in 1937 to minimize inventory and waiting times in production.

The continuous improvement method is closely associated with the Toyota Production System (TPS), which established the Lean concept as part of its strategy. Based on waste elimination, Lean aims to increase customer value and improve performance [8]. After studying the Ford technique, a group of Japanese engineers led by Shigeo Shingo and Taiichi Ohno, alongside Sakichi Toyoda, created the TPS. Toyota became the global leader in the industry by putting their thoughts into practice [9]. James Womack and Daniel Jones introduced Lean in their book "The Machine That Changed the World" in 1990 [10]. This method had a strong influence on Western businesses across various industrial sectors and produced outstanding results.

2.2 Lean Overview

According to Bhamu, the concept of "Lean" emerged in Japan after World War II [11]. Toyota's experiments gave rise to the Toyota Production System (TPS), built upon Fordism and Taylorism. Lean, which means "thin," is an approach and a mindset derived from Toyota's experiences. The TPS is an integrated socio-technical system developed by the Japanese automaker Toyota during the 1973 oil crisis, encompassing its philosophy and management practices, and serves as the formal application of Toyota-ism within Toyota [12]. The TPS manages a manufacturer's production and logistics, as well as interactions with suppliers and customers [13]. Over the decades, a significant amount of trial and error has contributed to the enrichment of the knowledge base, eventually leading to the development of the "Lean" concept [14, 15].

Lean is characterized by an open structure that allows each company to adapt and align methodologies according to its specific requirements and conditions. Organizations use the best Lean tools and practices to improve their supply chain management, which encompasses all aspects of the value chain from product creation to satisfying the end customer [16].

2.3 Lean Six Sigma Overview

Lean Six Sigma consists of two main approaches: Lean and Six Sigma. The Six Sigma approach was created by Motorola in 1980, and General Electric followed in 1990. It is a method aimed at eliminating manufacturing defects using statistical calculations [17]. It is a profit-maximization strategy used to meet customer demands. In 1986, the cellular phone company "Motorola" successfully implemented Six Sigma, and since then, many companies have followed suit [1]. Six Sigma has made a substantial contribution to improving the quality and productivity of businesses [18].

According to Antony, combining either of these two approaches, such as Lean or Six Sigma, can contribute to improving industrial outcomes [19]. Indeed, these two approaches have been identified as among the most significant factors in achieving continuous improvement. LSS, which stands for "Lean Zero Defects," is a robust strategy focused on customer satisfaction and based on the DMAIC approach (Define, Measure, Analyze, Improve, Control). According to Albliwi, LSS is a structured management method aimed at improving process quality and efficiency [20]. Initially, LSS was applied to industrial processes, but since then, it has been extended to all administrative, logistical, banking, commercial, and public processes [21]. Currently,
LSS is considered the latest generation of improvement methods. It is a rigorous improvement methodology that aims to enhance customer satisfaction, economic performance of the company, and achieve the company’s strategic objectives [17].

3. Problematic

Improving performance is an ongoing task to achieve the best results while considering all the economic and social elements of the company [22]. On the other hand, most of a company’s strategy focuses on the technical aspect to minimize waste and produce better, faster, and at a lower cost. As a result, many companies overlook social issues and the influence of reform efforts on them [4]. They rely on the concept of an economic performance system that assesses quality, cost, and time in three dimensions (Figure 1).

The companies and authors attach great importance to economic success while neglecting social considerations. The deterioration of these elements directly affects employee performance, including demoralization and demotivation of workers [11, 22]. Indeed, poor working conditions have a negative impact on economic and technical variables [9]. This conflict of interest between technical and social objectives creates an imbalance within companies and has an influence on the long-term viability of continuous improvement programs.

Therefore, a new idea is needed to control the influence of improvement initiatives on working conditions (Figure 2). The motivation and behavior of individuals are necessary for the successful integration of an improvement project [13]. That is why we have decided to broaden our research scope and propose a paradigm that considers social and technological concerns [23]. The objective of this project is to emphasize the importance of safety and ergonomics.

Moreover, the implementation of Lean has been highly successful in revolutionizing the quality, cost, and delivery time of businesses, making it a widely adopted method worldwide [24, 25]. Despite the high demand for this method over the past 30 years in various industries, complaints have been raised regarding the negative effects of Lean, particularly on factors such as employee safety, health, and motivation [22].

To further explore this topic, we initially conducted a literature review to gather more data on these approaches, followed by a professional survey conducted in a multinational company in the automotive sector. Based on a detailed literature review of Lean, Six Sigma, and LSS approaches, it was observed that researchers primarily focus on the technical aspect of these methodologies [4]. However, despite the technical and economic improvements, several practices within these approaches have a negative impact on the operational excellence of companies (Figure 3). This is attributed to the disregard for ergonomic, safety, and environmental factors in their development projects. Indeed, these aspects are often neglected, which can compromise the sustainability and overall performance of the company.
In recent decades, numerous complaints have been identified regarding the negative effects of Lean, particularly on employee safety, health, and motivation factors [26]. Therefore, our research work was conducted, initially based on this literature review feedback, to gather more data on these approaches. Subsequently, we conducted a professional survey of a multinational automotive company.

4. Professional Survey

We conducted a descriptive survey on the impact of the Lean method on economic and social factors (QCD and ESE) in the context of research on the integration of this method. The survey was conducted in a large company in the industry. This company is a subsidiary of a global conglomerate that has made continuous improvement a priority in its manufacturing system since 2010. It specializes in the production of cars for export to over 70 countries. Figure 4 illustrates the method used to conduct this survey in a professional context. After several weeks of preparation in collaboration with consultants and industry experts, we developed a questionnaire that we administered to senior and middle-level executives, as well as employees and workers responsible for machine supervision.

This survey is a crucial moment in our study and constitutes an essential part of conducting our case study. On one hand, we will be able to utilize the results of our analysis to determine the constraints of implementing Lean. On the other hand, it will allow us to identify the barriers to integrating the Lean approach in the automotive sector of developing countries. Finally, this work provides us with a unique opportunity to bridge the gap between literature research and the practical insights from operators, who represent the reality on the ground.

Our study focuses on the survey results conducted within this company, which will enable us to evaluate the impact of continuous improvement on ESE factors and measure employee satisfaction regarding the changes brought about by Lean or LSS projects.

This paragraph describes the various phases of the survey process. Firstly, the questionnaire developed for this survey consisted of three parts: The first part of the survey aimed to gather general information about the company and the department or activity area. The second part aimed to obtain the profile of the respondents, such as their educational background, Lean practice, job function, experience, etc. The third part aimed to collect data on the benefits and drawbacks of Lean implementation, and to identify the impacts of these initiatives on QCDESE factors.
4.1 Summary and Results

The results of the questionnaire evaluations from 150 individuals out of a total of 252 respondents in this survey are listed below. This corresponds to a response rate of 59%. According to Table 1, the results of this study suggest that 71.3% of individuals believe that Lean & LSS degrade the social SE elements, while 28.7% believe that there is no degradation. In comparison to managers, we also observe a significant ratio of workers and employees since, in reality, it is the workers who undergo Lean & LSS transformations. These results suggest that Lean & LSS practices have a detrimental influence on working conditions.

<table>
<thead>
<tr>
<th>Table 1. Survey Data Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effect of Lean</td>
</tr>
<tr>
<td>Total</td>
</tr>
<tr>
<td>Managers</td>
</tr>
<tr>
<td>Employers</td>
</tr>
<tr>
<td>Workers</td>
</tr>
<tr>
<td>Total</td>
</tr>
</tbody>
</table>

This survey conducted as part of our project has provided us with valuable information on the impact of adopting Lean and LSS on social factors. Through this survey, we were able to assess employees' perceptions of the changes implemented within the company and their feelings towards these changes. This survey, which we conducted in this project, allowed us to gather valuable information on the impact of adopting Lean and LSS on social and environmental factors. Through this survey, we were able to assess employees' perceptions of the changes implemented within the company, as well as their feelings toward these changes.

On the one hand, this information is essential to provide operational employees with a concrete vision of the effects of continuous improvement methodologies. It also allows us to measure the effectiveness of Lean and LSS projects in terms of safety and ergonomics. On the other hand, the results of this survey confirmed the presence of risks of degradation of working conditions, based on the data collected from employees. The comparison of this data with the literature synthesis also demonstrated the relevance of the research results for the studied company.

4.2 Discussion

The study of Lean & LSS implementation reveals that these practices can lead to significant changes in work organization. These modifications can impact the number of tasks to be performed, the restructuring of roles and responsibilities, the emergence of work-related stress, and the redefinition of tasks for each department or position. It is important to note that the introduction of new challenging tasks or increased responsibilities can elevate employee stress levels, which can have both positive and negative repercussions. Indeed, the level of stress is a critical factor that directly impacts employee motivation, engagement, and the overall work environment. Improving the work environment, as well as respecting the culture and organizational structure, has a significant positive impact on the quality of work life [22]. As a result, companies must ensure that:

- Improving working conditions is embedded in their overarching strategy.
- All employees receive basic training in continuous improvement.
- The Lean/LSS projects should be led by expert pilots who ensure the consideration of SE elements.
- The commitment of the management in Lean & LSS projects.

Therefore, the current approaches and measures are no longer sufficient to guarantee these objectives, and there is no means to measure and track the progress of these SE factors. As a result, we have proposed a new concept called Lean Management Plus (LMP) to prioritize the social and organizational aspects in future changes or continuous improvement. LMP builds upon the standard Lean Management approach while considering other dimensions, such as the safety and ergonomics of our personnel. Indeed, improving the quality of work life in the professional world is increasingly demanded by social partners.

In summary, we examined the Lean/Kaizen initiatives that had been implemented in the observed departments, and we collected feedback from participants regarding the impact of these projects on the SE variables. In recent years, Morocco has made numerous attempts to integrate Lean principles into the Moroccan industry:

- In 2011, Mohammed VI launched the first manufacturing model factory in Africa and the Middle East, "INMAA: a model tested in many Western countries."
The Moroccan government implemented an industrial acceleration strategy for the years 2014 to 2020, aiming to foster high-performance ecosystems.

Morocco joined the German project "Compact with Africa" in 2017.

Morocco implemented a new flexible exchange rate system for the dirham in January 2018.

All these changes aim to enhance the attractiveness and competitiveness of the Moroccan economy by establishing an efficient production system that promotes value development and intangible capital investment, with the goal of joining the ranks of emerging countries by 2040.

5. Introduction of the New LMP Model

Lean Management Plus (LMP) is a novel model developed within the framework of production management improvement to optimize performance while minimizing the risks of social degradation. It is considered an evolution of Lean to address social constraints that may hinder the success of this approach. Indeed, Lean has often been criticized for its excessive focus on economic efficiency at the expense of worker safety and ergonomics (Figure 5). LMP, therefore, considers the interactions between different processes and the individuals involved in them, aiming to optimize the entire system rather than focusing on a single aspect.

The LMP approach has introduced innovation by creating a new performance indicator called OEEP to monitor the evolution of economic and social performance indicators, thus avoiding the risk of improving one factor at the expense of another. This approach provides an overview of the impact of each continuous improvement action on the entire system and enables informed decision-making based on the results obtained. By using OEEP, companies can track their performance evolution across all economic and social aspects and take necessary measures to address any gaps. This ensures overall and sustainable continuous improvement of the company's performance while preserving the health and well-being of workers.

OEEP is a more comprehensive dashboard than the traditional Total Synthetic Performance Rate (OEE) as it enables deviation tracking of the targets set by decision-makers.

6. Implementation of LMP

LMP is applicable in a Kaizen context with a focus on motivating employees and respecting organizational and cultural aspects. It utilizes Lean tools to enhance our industrial system by following the philosophy and principles of Lean Management to achieve perfection, improve quality, and eliminate waste in a supportive and motivating work environment. LMP is a structured approach that has been developed to study the potential benefits of integrating safety, ergonomics, and environment in any type of change or improvement project. LMP oversees a given process to enable companies to progress toward perfection and improve quality while eliminating waste [22].

Here are the key steps for implementing the LMP approach:

Step 1  Assess current performance indicators, including data related to ergonomics, safety, and the environment.

Step 2  Identify the QCDESE factors with discrepancies and constraints encountered in the field, using the initial OEEP calculation as the primary indicator for tracking the evolution of these factors.

Step 3  Identify performance tracks and actions to be implemented in order to achieve the set objectives, while measuring the evolution of the OEEP.

Step 4  Implement measures such as coaching, training, and testing to ensure the achievement of the planned objectives in the Lean project.
Step 5  Monitor and verify the results obtained in the field by comparing the final OEEP with the initial OEEP.
Step 6  Finally, standardize and disseminate the identified best practices across the organization.

This concept appears to perfectly meet the needs of managers and researchers to address discrepancies and improve performance across all QCDESE factors.

7. Overall Equipment Effectiveness Plus OEEP

The Overall Equipment Efficiency (OEE) is the most well-known indicator in the industrial world. It provides a comprehensive reflection of the equipment's state on the production site and is widely used in the manufacturing industry to analyze equipment efficiency [27]. Its standard formula, given by Equation 1, is based on the three pillars: Quality, Cost, and Lead Time.

$$OEE = T_p \times T_p \times T_Q$$  \hspace{1cm} (1)

* $T_p$: Availability Rate;  
* $T_T$: Performance Rate;  
* $T_Q$: Quality Rate.

However, this indicator does not adequately consider the social and organizational aspects of a long and complex industrial chain. In order to involve workers and better protect them, it is relevant to monitor and take into account the ESE factors in the new indicator (OEEP). The OEEP represents the ratio between actual production and the maximum theoretical production (Equation 2).

$$OEEP = \frac{\text{Actual output}}{\text{Maximum possible output}}$$  \hspace{1cm} (2)

The functional relationship between OEEP and OEE is as follows (Equation 3):

$$OEEP = OEE \times T_E \times T_S \times T_E$$  \hspace{1cm} (3)

- $T_E$ is a real rate between 0 and 1, related to the Ergonomic of the personnel.
- $T_S$ is a rate between 0 and 1, related to Security.
- $T_E$ is a rate between 0 and 1, related to the Environment.

The OEEP is an indicator that measures the success of a project or production process. The OEEP is a comprehensive indicator for measuring the mastery of all types of changes. It consists of two parts:

- The well-known and widely used traditional OEE, which varies depending on the company.
- The Rates ($T_E$, $T_S$, $T_E$) that track the evolution of improvements and changes in a workstation.

To assess the Rates ($T_E$, $T_S$, $T_E$), we will use the FMECA tool (Failure Mode, Effects, and Criticality Analysis). This method identifies potential failure modes of a product, process, or procedure, which helps prevent problems before they occur. To evaluate the safety risk, ergonomics, or environmental impact, we will use the following formula (Equation 4) to calculate the criticality:

$$C = G \times F \times M \times S$$  \hspace{1cm} (4)

The evolution of $C$ related to the risk failure in the analysis is as follows: $C$: Criticality, $G$: Severity; $F$: Frequency; $M$: Control; $S$: Environmental Sensitivity (reserved for the environment).

The calculation of the rate for each risk i for any factor within the performance factors "ESE" is given by (Equation 5):

$$Z_i = \frac{M_i \times G_i \times F_i \times S_i}{M_i \times G_i \times F_i \times S_i}$$  \hspace{1cm} (5)

So, the compliance rate will be calculated using the following formula (Equation 6):

$$t_i = 1 - Z_i = 1 - \left(1 - \frac{M_i \times G_i \times F_i \times S_i}{M_i \times G_i \times F_i \times S_i}\right)$$  \hspace{1cm} (6)

However, the compliance rate for all identified risks "n" is the product of the compliance rates for all risks within a specific area (station, line, workshop...) under evaluation (Equation 7).

$$t_i = \prod_{i=1}^{n} (1 - Z_i)$$  \hspace{1cm} (7)

- "i" is the social factor to be calculated (E, S, Er).
- "I" represents the aspect, risk, or constraint being evaluated.
- "n" is the number of aspects, risks, or constraints under evaluation.

So (Equation 8):

$$T_{aux} (T_E, T_S, T_E) = \prod_{i=1}^{n} \left(1 - \frac{M_i \times G_i \times F_i \times S_i}{M_i \times G_i \times F_i \times S_i}\right) \times \prod_{i=1}^{n} \left(1 - \frac{M_i \times G_i \times F_i \times S_i}{M_i \times G_i \times F_i \times S_i}\right) \times \prod_{i=1}^{m} \left(1 - \frac{M_i \times G_i \times F_i \times S_i}{M_i \times G_i \times F_i \times S_i}\right)$$  \hspace{1cm} (8)

- $z$: the number of environmental aspects under study.
- $n$: the number of ergonomic constraints being evaluated.
- $m$: the number of identified safety risks.
OEEP is a key indicator for monitoring and evaluating the effectiveness of our actions. It also helps prioritize factors and select the appropriate Lean Management tool for conducting a detailed diagnosis.

8. Case Study

The inefficient use of continuous improvement tools has the potential to harm a company’s performance and make tool activities non-sustainable. This results in suboptimal social and organizational outcomes, which have a negative impact on working conditions and employee motivation. This is the case in this study, where a conflict between "Technical & Social" performance aspects has been identified. The following case study describes the social gaps and anomalies that emerged following the implementation of a Lean project in a Moroccan automotive company in 2022.

The main objectives of the project are to achieve a productivity level of 9 operators on this production line, as defined by the project management.

- A 40-minute gain for each work team is desired.
- Decrease the cost of fixing and storing production averages on the line (-25%).
- Enhance the 5S status of this manufacturing area.

Figure 6. Explanatory diagram of Lean-affected positions.

Though the economic objectives of this site were achieved in the short term, four workplaces experienced a deterioration in their working conditions due to ergonomic and safety issues, as shown in Figure 6. These issues have a significant impact on the long-term viability of the improvement project. Despite progress in the industry regarding standards and criteria for occupational health and safety management systems, the results in terms of accidents and occupational disorders remain unsatisfactory despite preventive tools and protective measures. Therefore, it is important to take action to improve working conditions and ensure the long-term viability of the project.

8.1. Implementation of (LMP, OEEP)

A group was formed to monitor the performance of this production chain, particularly regarding the ESE factors. The collected data helped identify problematic positions and critical operations that have a significant impact on working conditions. To find solutions for improving performance and better protecting the workers, a more thorough analysis was conducted. To do so, the group used Table 2 to calculate the initial ESE rates and gain a better understanding of problem areas. Subsequently, a brainstorming session was organized to identify the actions to be implemented for improving performance and protecting workers more effectively.

In summary, Table 2 has identified several risks that could endanger workers and lead to productivity losses, unexpected work stoppages, and a decrease in operator and technician motivation. These risks have had a significant impact on the overall performance of the production chain in terms of quality, costs, and lead times, as evidenced by the data obtained during the calculation of OEE and on-site observations.

Table 2. Calculation of the rates for Environment, Safety, and Ergonomics in the initial state.

<table>
<thead>
<tr>
<th>Risk or Impact</th>
<th>Nominal indices</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Environment</strong></td>
<td>G</td>
</tr>
<tr>
<td>Increase in energy bill: Natural resources</td>
<td>1</td>
</tr>
<tr>
<td>Pollution: Presence of dust and smoke</td>
<td>1</td>
</tr>
<tr>
<td>Regulatory non-compliance</td>
<td>1</td>
</tr>
<tr>
<td>Soil contamination (spillage of hazardous products)</td>
<td>1</td>
</tr>
<tr>
<td>Drought + increase in water prices</td>
<td>1</td>
</tr>
</tbody>
</table>
According to the preparatory measures of the LMP project, such as training, awareness-raising, and the integration of all stakeholders in the company, as well as the commitment of management from the beginning of the project, we observed that all employees were involved in implementing preventive or corrective actions to address the previously identified anomalies and risks. There was active employee participation in defining both technical and social objectives, which facilitated the smooth progress of different project phases, particularly in testing and validating actions for all stakeholders.

After identifying the main causes of anomalies in the production line, following the LMP method, we implemented corrective and preventive actions in our process. These actions had a positive impact on all factors, and team members concluded that ESE factors should be prioritized in LMP projects. This approach had a significant effect on employee engagement and greatly improved other QCD factors (Table 3). By involving the operational staff in this project, we can ensure the maintenance of their commitment.

| Safety |
|------------------|------------------|------------------|------------------|------------------|
| Electrical risks: Damaged and non-compliant wires and connectors | 3 | 2 | NC | 12 | 14.8% | 85.19% |
| Hazards related to moving equipment and machinery (vehicles) | 2 | 3 | 2 | NC | 12 | 14.8% | 85.19% |
| Slip hazards due to water presence & oil leaks | 2 | 2 | 2 | NC | 8 | 9.88% | 90.12% |
| Degraded and non-compliant tools pose risks of falling parts / packaging | 1 | 2 | 2 | NC | 4 | 4.94% | 95.06% |
| Risk of infection, allergies: Presence of particles | 1 | 1 | 3 | NC | 3 | 3.70% | 96.30% |
| Tripping or injury risk due to metal falls on the ground | 1 | 2 | 2 | NC | 4 | 4.94% | 95.06% |
| Professional fatigue and lower back pain | 2 | 1 | 1 | NC | 2 | 2.47% | 97.53% |
| Risk of visual problems leading to accidents | 2 | 2 | 2 | 3 | NC | 12 | 14.81% | 85.19% |
| Excessive movement > 14 m/min and handling weights > 20kg | 2 | 3 | 2 | NC | 12 | 14.81% | 85.19% |
| Noise emitted by the chain, compressors, tools, motors | 2 | 2 | 2 | NC | 8 | 9.88% | 90.12% |
| Pain related to poor posture (neck, back, and upper limbs) | 3 | 2 | 2 | NC | 12 | 14.81% | 85.19% |

8.2. Analysis and Results

According to the preparatory measures of the LMP project, such as training, awareness-raising, and the integration of all stakeholders in the company, as well as the commitment of management from the beginning of the project, we observed that all employees were involved in implementing preventive or corrective actions to address the previously identified anomalies and risks. There was active employee participation in defining both technical and social objectives, which facilitated the smooth progress of different project phases, particularly in testing and validating actions for all stakeholders.

After identifying the main causes of anomalies in the production line, following the LMP method, we implemented corrective and preventive actions in our process. These actions had a positive impact on all factors, and team members concluded that ESE factors should be prioritized in LMP projects. This approach had a significant effect on employee engagement and greatly improved other QCD factors (Table 3). By involving the operational staff in this project, we can ensure the maintenance of their commitment.

Table 3: Calculation of Traditional OEE and OEEP.

<table>
<thead>
<tr>
<th>Phase</th>
<th>OEE</th>
<th>OEEP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(T_e)</td>
<td>(T_a)</td>
</tr>
<tr>
<td>Initial</td>
<td>79.9%</td>
<td>81.9%</td>
</tr>
<tr>
<td>Final</td>
<td>97.7%</td>
<td>96.9%</td>
</tr>
</tbody>
</table>

Summary: We have achieved our objectives within a professional framework and with a team spirit, thereby meeting the agreed-upon deadlines with management. The LMP approach has been our guide to achieving our goals by integrating all QCDESE performance axes. We have improved the working conditions of our employees and created a motivating environment.

9. Conclusion

The implementation of Lean & LSS strategy in business improvement and transformation initiatives is strongly encouraged by academics. However, this does not exclude the involvement of all stakeholders in the company to achieve the project objectives. Both the personnel and the results will be necessary to ensure the long-term viability of the planned initiatives.

The Lean & LSS strategy is a continuous improvement technique that enables us to develop and evolve projects with local teams. The human element is at the core of many continuous improvement methods, and employee participation and motivation are powerful drivers of change. While continuous improvement is crucial to meet market requirements, it should not be pursued at the expense of social factors such as safety and ergonomics. It is important to consider these factors from the early stages of the continuous improvement process to ensure sustainable growth for the company and a satisfactory quality of life for workers.

However, scientific research has revealed certain limitations. For instance, the lack of real case studies in many domains of activity makes it challenging to identify authentic data and probable consequences of using Lean & LSS methodologies on social and cultural variables. Additionally, it is important to identify other social parameters (such as culture, organization, and activity) that can have a significant impact on the results achieved within the LMP project framework.

References

2. dos Santos, Z.G., Vieira, L., and Balbinotti, G. Lean Manufacturing and Ergonomic Working Conditions in the Automotive...


Adaptive Approach for Dynamic Spectrum Utilization in Wireless Communication System

Niranjan S. Kulkarni*, Sanjay L. Nalbalwar, and Anil B. Nandgaonkar

Department of Electronics and Telecommunications Engineering,
Dr. Babasaheb Ambedkar Technological University, Maharashtra, India

Abstract

The evolution of wireless communication with new architectures has led to faster and more reliable wireless communication. However, with the rapid increase in the demanded services and user interface, available resources are becoming constraints in providing high-Quality of Services (QoS). Very High-Frequency Land Mobile Radio (LMR) communication is used as a means of data exchange over long-range wireless communication. LMR is designed with spectrum-sharing capability for various resource constraint applications. The existing approach of spectrum sensing using an energy-based detection technique is the widely used method in spectrum sensing and allocation. Various previous methods defined for spectrum utilization are developed with an assumption of a linear varying channel model, however, the dynamic variation in user interface and varying channel interference develops a limitation in resource utilization under dynamic channel conditions. Learning methods developed in optimizing resource allocation observe a large processing overhead under dynamic conditions. Addressing the issue of dynamic communication conditions, this paper outlines a method for Adaptive learning of estimation parameters in spectrum sharing for the cognitive wireless communication system. The dynamic spectrum variation is monitored in the resource-sharing process for higher system performance. The observations for the developed method illustrate an increase in system throughput, less delay, and system overhead in the network.

Keywords: adaptive learning; dynamic resource allocation; cognitive wireless communication

© 2023 Totem Publisher, Inc. All rights reserved.

1. Introduction

The rapid evolution of communication architecture has resulted in new means of communication which have the capability of self-tuning, detection, updating, and decision-making. The existing communication system is getting constraints with the increase in new services and applications. The frequency range for high-frequency wireless communication is typically between 150-174MHz. However, due to limited frequency availability and the constant addition of new services, this range has become highly congested [1]. Existing LMRs are processed with a frequency division approach where the spectrums are permanently allocated to each user. This leads to spectrum constraints in the network [2,3]. It has a large impact on critical applications and hence a new channel of communication is needed. Hence, it is required to develop new methods for spectrum-sharing interfaces. Cognitive properties in Land Mobile Radio systems have been significantly used in recent works [4,5].

In a network with cognitive devices, the free spectrum of the primary user can be effectively used by an opportunistic user that has cognitive radio capabilities. The free spectrum sensing approach was developed in various past works [6-10]. The author [11] has implemented a spectrum sensing approach to GNU radio for sensing and frequency modulations that was used for the transmission of data. However, the application is limited to frequency modulation which is less suited for high-frequency application. In developing an optimal solution in estimation, deep learning is presented in [12]. This method interfaces different layers of data-driven interlinked neural networks for frame-wise signal processing. The framing results in faster processing in a linear channel model. In [13] application of machine learning on spectrum utilization of visible light communication is presented. This approach developed a converging method for delay and throughput in visible light communication. The system optimizes the queue allocation by minimizing the data buffering, resulting in faster processing. In [14] an automatic identification system for a wireless communication network is proposed. This approach developed a

* Corresponding author.
E-mail address: niranjansk183@gmail.com
method for spectrum sensing without interfering with simultaneous services in communication. The simulation model is developed and illustrates the analysis of the spectrum utilization in detail. However, the system is limited to the simulation model only. No real-time test has been developed. The system focused on the detection of free spectrum in a bound frequency range only.

In [15] a low-cost device for spectrum monitoring using Dragon board 410C and RTL-SDR 2832U dongle is used for sensing free spectrum in the TV frequency range. The proposed system performs a scanning operation for used and non-used spectrums in communication. Spectrum sensing is limited with the spectrum sensing option for free spectrum where redundant spectrums are not been sensed. In developing LMR-based communication in the high-frequency range, GNU radio was used for real-time spectrum sensing and utilization operation. This approach performs the spectrum sensing operation based on the energy detection technique [16]. The estimation of free spectrum based on energy detection was observed in various applications [17,18], where the energy of the received signal is measured as a parameter for sensing free spectrum for communication. In [19], the developments in the area of machine learning for wireless communication are reviewed. The block-based processing in the deep learning approach has shown a better estimation and resource-sharing performance in wireless communication. The energy detection approach is an optimal solution in the area of spectrum sensing and is widely used in various communications, however, the existing energy detection approach for free spectrum sensing in the high-frequency range [20] is developed for a linear channel model. Under dynamic channel conditions, the energy detection needs to be adaptive to attain a better sensing performance.

In developing a dynamic sensing model, this work outlines a method of dynamic energy detection approach for spectrum sensing in wireless communication. To present the outlined method, this paper presents the existing approach of spectrum sensing in Section 2. Section 3 describes the proposed approach of spectrum sensing and allocation based on dual monitoring parameters. Section 4 presents the simulation result for the developed approach and Section 5 concludes the proposed work.

2. Resource Utilization and Decision System in Wireless Communication

In optimal spectrum utilization, spectrum sensing and sharing is an optimal solution in spectrum scare communications. Spectrum sensing is a critical part of the spectrum-sharing approach as the method defines the spectrum sensing based on the signal characteristic for free spectrum decision-making. In developing methods for spectrum sensing, the energy-based approach is the most used method. The energy-based approach is chosen for its low computational cost, ease of operation, and simple method for energy detection. The method is also used in the detection of user engagement in the transmission process. In a real-time interface, the sensing operation is made on SDR to detect the primary user’s engagement in transmission. The estimation is developed based on two-level probabilities, shown in Equation 1,

\[ P_{r1} : R_{xi} = n_i \]
\[ P_{r2} : R_{xj} = S_i + n_j \]  

(1)

where, \( S_i \) is the transmitted signal in the channel, \( R_{xi} \) is the received signal at the receiver passed to the estimator, and \( n_i \) is the channel interference noise of additive white Gaussian noise type with identical and independent distribution with zero mean. The two probabilities state the case of presence of only noise or signal with noise. The energy of the received signal is computed as Equation 2.

\[ E = \sum_{k=1}^{m} |R_k(k)|^2 \]  

(2)

The signal is processed for 1 to m coefficient count. In making a decision for spectrum sensing, the detection is performed by a threshold observing signal with the learned reference, where a Constant False Alarm Rate (CFAR) is used and defined as Equation 3 [21].

\[ CFAR = G \left( \frac{th \sigma^2 - k}{\sqrt{2k}} \right) \]  

(3)

Here, \( G \) is the standard Gaussian complementary Cumulative Distribution Function (CDF) and \( \sigma^2 \) is the noise variance processed for \( k \) number of samples, and \( th \) is the energy threshold value. The system detection probability is defined as Equation 4.
The estimation of the threshold is derived as Equation 5, which results in Equation 6.

\[ G^{-1} (\text{CFAR}) = G^{-1} \left( \frac{\theta h}{\sigma^2 - k} \right) \]

\[ \theta h = \left[ G^{-1} \left( \text{CFAR} \times \sqrt{2k} \right) + k \right] \times \sigma^2 \] (6)

The estimation process is developed based on the computed energy, which is the averaged square sum of all received signal coefficients, given as Equation 7.

\[ E = \frac{1}{m} \sum_{k=1}^{m} | R_i (k) |^2 \] (7)

In the estimation process the energy derived from the received signal is compared with the threshold value in making decision of primary user's engagement. The selection process is illustrated in Equation 8.

\[ D = \begin{cases} 
E > \theta h \Rightarrow P_{r_1} \\
E < \theta h \Rightarrow P_{r_2} 
\end{cases} \] (8)

where, the estimated energy greater than the threshold is user engagement in communication. If the computed energy is lower than the threshold the spectrum is sensed as free and can be shared for communication. However, the signal energies are computed on the assumption of a linear channel model. In current communication, the channels are dynamically varying and the linear model needs to be updated for a dynamic condition.

3. Adaptive Decision Method for Estimation and Sensing Process

The conventional approach of energy detection, developed based on a linear channel model, is constrained for estimation under limited conditions. In the present communication, the dynamic nature of the channel needs a modified approach of estimation which should be adaptive for estimation under varying interference conditions.

In the estimation of energy, the threshold of the system is developed as a function of signal characteristics which are defined by the shape and scaling parameter. In order to determine if a received signal is above or below a threshold, a probability function for estimation is used. This function is given as Equations 9 and 10,

\[ P_f = P\{E > \theta h \mid P_{r_1}\} \] (9)

\[ P_d = P\{E > \theta h \mid P_{r_2}\} \] (10)

where \( P_f \) and \( P_d \) are the probability of false alarm and probability of detection.

The estimation of the threshold under dynamic channel conditions is developed as a probabilistic function \( P(\cdot) \) where the threshold \( (\theta h') \) is defined by Equation 11, and defined as an inverse function of the estimation for the probability and is the shape and scale parameters of the received signal respectively.

\[ \theta h' = F_{E[P_f]}^{-1} \left( 1 - P_f, Sc, Sh \right) \] (11)

Energy detection can be maximized with the generalized likelihood function which is defined by Equation 12.

\[ E = \frac{1}{n} \sum_{k=1}^{n} \left( \frac{| R_i (k) |}{\sigma^2} \right)^p \] (12)

where the energy computation is tuned to an amplifying parameter \( p' \) for maximum energy detection. The parameter \( p' \) is varied to attain a maximization of energy for varying noise levels and the decision of the proposed approach is modified as Equation 13.
The process of estimation for the free spectrum is outlined in the flow chart illustrated in Figure 1 below.

\[
D = \begin{cases} 
E > \text{th} & \Rightarrow P_{r1} \\
E < \text{th} & \Rightarrow P_{r2} 
\end{cases}
\]  
(13)

In cognitive wireless communication the spectrum is shared by the PU to SU based on spectrum availability. The spectrum utilization has an impact on the communication performance of these networks. In this, the dynamic access of the user has a dynamic impact on the spectrum availability. The availability of the spectrum at any instance in cognitive wireless communication is given as Equation 14.

\[
S_{av1}^t = \sum_{n \in N} S_n^t - S_i^t
\]  
(14)

Where \( \sum_{n \in N} S_n^t \) is the total spectrum for all ‘n’ sensed nodes in the network with node ‘N’. The notation \( S_i^t \) illustrates the allocated spectrum for communication at a time ‘t’ for the \( i \)th user.

An additional spectrum is observed due to the dynamic access of user, which is given by Equation 15.

\[
S_{av1+}^t = \sum_{n \in N} S_n^t + S_{av1+}^t
\]  
(15)

The allocation was based on the synchronization using the full search or random search in the existing approach. The allocation, however, observes to have a higher searching time due to additional channels and introduces a higher interference. An illustration of the process of interference monitoring is shown in Figure 2.
In the proposed approach the channel allocation is governed by the interference due to additional spectrum. To optimize the spectrum allocation, which is given as Equation 16.

$$S'_{\text{alloc}} = \min_{\{n\}} \{I\} \Rightarrow S'_{\text{alloc}} \leq S'_{\text{alloc}}$$

Where the spectrum allocation is a constraint to minimum interference due to $n$ sensed users being subjected to the available spectrum less than the total spectrum. For each sensed spectrum interference is computed as the mean square error (MSE) between the actual and loop back signal from broadcast users. The interference is computed as Equation 17.

$$I_j = \text{mean}(\{(S_i - S_{ib})^2\})$$

The minimum of $I$ is considered as the convergence for spectrum allocation. This allocation constrains the spectrum allocation with minimum interference with maximum spectrum allocation. The proposed approach is offering optimal spectrum utilization with higher accuracy. In developing spectrum sensing, the energy-based method is used. The signal energy is dependent on channel interference. The channel interference is observed to be dynamic in nature, which limits the estimation performance in resource allocation. The learning characteristic for resource utilization was developed in various methods in the recent past. Spatial and temporal domain processing was developed in traffic prediction and resource utilization using cross-domain neural networks [22].

In the learning process, multiple data learning datasets were used in the classification of the traffic model. Different types of datasets define multiple traffics such as SMS, cell, and internet, which are observed for cross-domain and non-cross-domain traffic type. Resource utilization using spectrum allocation is proposed where the learning error is governed in the learning characteristic for resource utilization was developed in various methods in the recent past. Spatial and temporal domain processing was developed in traffic prediction and resource utilization using cross-domain neural networks [22].

In the learning process, multiple data learning datasets were used in the classification of the traffic model. Different types of datasets define multiple traffics such as SMS, cell, and internet, which are observed for cross-domain and non-cross-domain traffic type. Resource utilization using spectrum allocation is proposed where the learning error is governed in the learning characteristic for resource utilization was developed in various methods in the recent past. Spatial and temporal domain processing was developed in traffic prediction and resource utilization using cross-domain neural networks [22].

In developing the learning signal for decision reference signals were derived from past learning, which forms a dataset in decision making. In the learning system, past learning is used for making the current decision. In the presented work, a dataset of 1000 randomly trained information for different traffic conditions is narrated. The process of dataset generation is developed in 2 operational states. The first state a signal is generated from the transmission unit and passed to the channel model.

The second phase of the work varies the channel parameters and the signal variance at the receiver and is recorded. The process is iterated for 50 different data rates with 20 random variations. The process gave a range of observations for different channel conditions, which are used as learning information for the decision system. The estimation process updates the testing signal and process for the estimation. The process of weight updating is defined as Equation 19.

$$w^x = \left(X^{(x-1)}\right)^2 + \varepsilon_n$$

The existing method performs a convergence of estimation based on estimated error with weight update. The minimization of estimation error is developed as the SNR probability of the $n^{th}$ user given as Equation 20.

$$\varepsilon_n = \frac{1}{R} \exp \left(-\frac{\phi}{\phi}\right)$$

Where $\phi$ defines the current SNR of $n^{th}$ channel and $\bar{\phi}$ represent the mean SNR of $n^{th}$ channel where the instantaneous channel SNR is given by Equation 21.

$$\phi = \frac{|S|^2}{|N|^2}$$

Where $|S|^2$ and $|N|^2$ are the signal and noise strength computed each iteration. The estimation process converges with the satisfying of the minimum error constraint. The estimated signal is then processed for energy detection in making decisions of spectrum sensing and allocation. The estimation process improves the estimation due to minimal error criterion and hence results in increasing the system performance as observed in the following section.
4. Results and Discussion

The evaluation of the proposed system is made for a cognitive wireless communication system following spectrum sensing through which an energy detection model is developed. The performance of the system developed in the present work is compared with the performance of the system developed [1].

In the implementation, three base stations are considered with random distribution. Pseudo-code sequences are applied for spectrum spreading. The user data is transmitted over a Gaussian channel with fading effect. For the process of transmission, the value of the pulse duration is set to 2 ns, whereas the additional time shift values are set to 100 ns. A second-order Gaussian monocycle is used. In order to avoid the inter-symbol interference, the bit interval is set to 200 ns. For testing the performance, the number of data bits is taken as 1000 and the number of channels is taken as 250. The network implemented in the analysis of the presented work is shown in Figure 3.

![Network distribution](image)

Figure 3. Network distribution for the simulation with node placement

The simulated network is defined for an area of 30×30 m² with nodes randomly placed in the network area. The nodes are interlinked with the distance measured between the two nodes defined by Equation 22.

\[
\text{Dist}_\text{Node} = \sqrt{(i_2 - i_1)^2 + (j_2 - j_1)^2}
\]  

(22)

Where i, j are the coordinates of the node. To link the nodes the range constraint is monitored and all the nodes within a given range are defined as links. In monitoring the node inter-links, the nodes with maximum coverage are declared as the monitoring node.

The process of spectrum sensing is developed based on user engagement in communication. In sensing operation, the interference is minimized by the proposed error-based optimization and dynamic threshold method. In the presented work the complexity of the process is measured in terms of spectrum sensing probability and channel utilization. The system complexity is developed as a measure of the processing effort made in the estimation process. Due to an adaptive mean of resource governance, the effort made in estimation and decision-making gets faster. The dynamic range of learning and the iterative error minimization reduces the estimation overhead in terms of the number of iterations taken in the convergence of decision criteria. Spectrum sensing probability is observed to improve with the lower processing complexity in terms of varying input data load.

Wherein, existing approaches were developed with spectrum sensing based on the energy detected from received signals, the proposed approach estimates the energy parameter using the learning method. This developed a more accurate estimation of signal energy under dynamically varying channels, which reduces the overall convergence time and hence reduces the system complexity. The lower system complexity results in an increase in spectrum sensing probability and channel utilization as observed from the obtained results. The spectrum sensing probability of the network is defined as Equation 23.

\[
S_{\text{sense}} = \sum_{k=1}^{t} \frac{S_{\text{sense},k}}{S_{\text{total}}}
\]  

(23)
Where $S_{\text{sense}}$ is the sensed spectrum and $S_{\text{total}}$ is the total free spectrum available in the network observed for a period of ‘t’. The network throughput is measured as a ratio of communicated data over received data for an observing time, given as Equation 24.

$$\text{Thrp} = \frac{I - I'}{I} \times t$$  \hspace{1cm} (24)

Where $I$ actual data is transmitted and $I'$ is received data in the period t. To observe the utilization performance channel, utilization is governed, which is given as Equation 25.

$$\text{Ch\_util} = \frac{S_{\text{sense}}}{S_{\text{total}}} \times t$$  \hspace{1cm} (25)

Observations of the developed system in comparison with the existing sensing method using a self-organized method are made. The spectrum sensing probability for the proposed method is observed for varying interfacing channels. The interfacing channel is defined as the total number of free accessing channels in the network. It is observed that with the increase in the number of interfacing channels, the sensing probability is comparatively higher for the proposed method compared to the random access and parallel sensing methods as given in Table 1. The probability is observed to improve from 2-30 % with the increases in interfacing channel values. With the varying range of learning, the decision system was able to predict the free spectrum more accurately by categorizing the signal into noised or signal data.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>11</td>
<td>11</td>
<td>13</td>
</tr>
<tr>
<td>30</td>
<td>14</td>
<td>16</td>
<td>25</td>
</tr>
<tr>
<td>40</td>
<td>19</td>
<td>21</td>
<td>41</td>
</tr>
<tr>
<td>50</td>
<td>24</td>
<td>26</td>
<td>51</td>
</tr>
<tr>
<td>60</td>
<td>26</td>
<td>31</td>
<td>61</td>
</tr>
</tbody>
</table>

The existing system using parallel estimation [3] and resource allocation [1] is compared with the presented method. The spectrum sensing probability of the simulated system is shown in Figure 4. The categorization improves the energy detection and hence the spectrum sensing probability. Wherein, existing approaches have constraints in estimation with higher channel accessing due to false classification, the proposed method has a better classification due to more accurate learning operation. An increase in the free spectrum sensing improves the system throughput as more data is accessed in the same time interval. The system throughput to the presented network is shown in Figure 5. The throughput is improved by the utilization of spectrum sensing and allocation method in an adaptive manner which resettle in higher traffic flow.
Adaptive Approach for Dynamic Spectrum Utilization in Wireless Communication System

giving an increase in system throughput.

With the increase in free channel sensing, the observed throughput increases as the data accessing probability gets improved with the availability of more free channels. Wherein the detection of free spectrum is lower in the existing method, which is more separated at higher free channels the data exchange gets reduced, minimizing the system throughput. For the proposed method the throughput is observed to improve from 0.3 to $1.1 \times 10^9$ bps for 20 to 60 count of free channel availability. The increase in the system throughput reduces the system overhead by processing more volume of data and hence results in lower processing complexity to the system. Observation for the system throughput of the testing network is listed in Table 2.

The increase in spectrum sensing probability increases the channel utilization, which is observed from Figure 6 below. Channel utilization shows the performance of system accuracy in sensing and utilization of free spectrum which improved the overall system throughput as observed in Table 2.

From Figure 6 it is observed that for a higher free channel count, the system resource allocation is observed to be more optimally used. The observations for the presented simulation listed in Table 3 illustrate the channel utilization for the proposed method has an increase in the count from 9 to 21 compared to the parallel sensing method and 14 to 37 count for the random sensing method.

The observed parameters are effective with the input data rate. Input data rate defines the volume of data interfacing in transmission which also defines different service interfaces into the network. The increasing data rate offers a larger overhead of processing data which impacts the system’s complexity and estimation performance. The observed parameters of spectrum sensing probability is listed in Table 4, network throughput is listed in Table 5, and channel utilization is listed in Table 6 with varying data rates shown in Figure 7-9 respectively.

![Figure 6. Channel utilization with change in channel metric](image)

![Figure 7. Spectrum sensing with varying input data](image)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>6</td>
<td>11</td>
<td>20</td>
</tr>
<tr>
<td>30</td>
<td>9</td>
<td>16</td>
<td>34</td>
</tr>
<tr>
<td>40</td>
<td>11</td>
<td>21</td>
<td>32</td>
</tr>
<tr>
<td>50</td>
<td>14</td>
<td>26</td>
<td>44</td>
</tr>
<tr>
<td>60</td>
<td>16</td>
<td>31</td>
<td>53</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>60</td>
<td>88</td>
<td>95</td>
</tr>
<tr>
<td>3</td>
<td>46</td>
<td>70</td>
<td>75</td>
</tr>
<tr>
<td>5</td>
<td>33</td>
<td>50</td>
<td>55</td>
</tr>
<tr>
<td>7</td>
<td>20</td>
<td>30</td>
<td>35</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>10</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.4</td>
<td>2.6</td>
<td>3.9</td>
</tr>
<tr>
<td>3</td>
<td>1.38</td>
<td>2.58</td>
<td>3.3</td>
</tr>
<tr>
<td>5</td>
<td>1.2</td>
<td>2.3</td>
<td>2.8</td>
</tr>
<tr>
<td>7</td>
<td>0.7</td>
<td>1.48</td>
<td>2.3</td>
</tr>
<tr>
<td>9</td>
<td>0.3</td>
<td>0.5</td>
<td>1.9</td>
</tr>
</tbody>
</table>
Table 6. Channel utilization in the network

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28</td>
<td>55</td>
<td>78</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>54</td>
<td>67</td>
</tr>
<tr>
<td>5</td>
<td>23</td>
<td>46</td>
<td>57</td>
</tr>
<tr>
<td>7</td>
<td>16</td>
<td>30</td>
<td>47</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>10</td>
<td>39</td>
</tr>
</tbody>
</table>

The increase in input data rates increases the volume of data engaged in accessing and hence increases the probability of false classification. The increase in false detection reduces the spectrum sensing probability. However, an increase in the learning data and proper classification results in higher sensing probability as shown in Figure 7. In the presented network the sensing probity for the proposed work is improved by 60-95% for the random sensing method and 88-95% for the parallel sensing method.

The improvement in the spectrum sensing results in a higher system throughput which is observed in Figure 8. Observation for the system throughput and channel utilization shown in Figures 8 and 9 illustrates the increase in the system performance under an increase in interfacing data rate. The system throughput and channel utilization are observed to decrease with the increase in interfacing data rate as the volume of data for accessing increases and lower channel utilization blocks the data from exchanging resulting in lower throughput.

A higher channel sensing is due to accurate learning, resulting in more sensing probability, which increases the utilization performance and hence increases the system throughput. For the testing network, the system throughput is observed to increase from 1.4 to 3.9×10^9 bps and 2.6 to 3.9×10^9 bps for the proposed method compared to the random access and parallel sensing method. At a data rate of 9 Kbps, the network shows an increase in throughput from 0.3 to 1.9×10^9 bps and 0.5 to 1.9×10^9 bps for the proposed method compared to the random and parallel sensing methods respectively. The channel utilization at 1kbps is observed to increase from 28 to 78 channel counts and 55 to 78 counts for the proposed method. At a higher data rate of 9 kbps, the utilization count increases from 6 to 39 and 10 to 39 for the proposed method compared to the existing random and parallel method.

5. Conclusion

This paper outlined a resource-sharing approach for cognitive wireless communication in land mobile radio (LMR) communication using an adaptive learning approach. A method is outlined for dynamic energy detection in spectrum sharing under very high-frequency communication. A novel approach to spectrum allocation is also introduced for optimal utilization of sharing spectrum. The spectrum holes are detected using a modified energy detection approach and the sensed spectrum is controlled by the monitoring of the interface in the channel. The presented approach results in more efficient spectrum utilization by the convergence of signal estimation using an error minimization technique. The learning approach developed minimizes the error probability, hence an accurate decision system resulting in higher accuracy of signal classification. The increase in classification performance results in an increase in system throughput which is observed to increase by 30% for the proposed method. The spectrum sensing probability is increased by 40% and channel utilization is improved by 41%. For varying interfacing data rates, the spectrum sensing is increased to 7%, system throughput increased...
to 33%, and channel utilization to 29%. The proper learning data interface, weight updating, and efficient allocation increased the channel sensing and utilization performance. The estimation could further be improved by the realignment and clustering of learning data which is a future scope of the presented work.

References


An Advanced Machine Learning Approach for Student Placement Prediction and Analysis

K. Eswara Rao\textsuperscript{a,}\textsuperscript{*}, Bala Murali Pydi\textsuperscript{b}, T. Panduranga Vital\textsuperscript{a}, P. Annan Naidu\textsuperscript{a}, U. D. Prasann\textsuperscript{a}, and T. Ravikumar\textsuperscript{a}

\textsuperscript{a}Department of Computer Science and Engineering, Aditya Institute of Technology and Management, Tekkali, India
\textsuperscript{b}Department of Electrical and Electronics Engineering, Aditya Institute of Technology and Management, Tekkali, India

Abstract

As there are job opportunities worldwide, the graduates who are being produced in large numbers from various backgrounds are constantly trying to get them. Moreover, the management of graduate colleges gives proper training to the students to get those opportunities. Every student has their skills, unique creative outlook, studying, and good academic skills that help them get placed in various companies and also have a chance to get reputed positions, but most of the graduates are still failing to get the opportunity because they cannot find what skills to acquire. For this reason, in this paper, we gathered information from students who have finished their courses at different colleges. Collected information by communication and asked them about their academics, performance, families, skills, personal information, habits, etc., and what prevented them from taking the opportunity. Then, we made a dataset with all the factors that affected a student's career and used that to create a model with synthetic data. Student Placement Prediction can also benefit colleges and universities by providing valuable observations of student career outcomes. By understanding the factors influencing student job placement, colleges can conduct services and programs to help their students be better prepared for their careers. Accuracy and precision were used to evaluate the eXtreme Gradient Boost (XGBoost) machine learning model's performance compared to standard classification techniques. According to the results, the proposed algorithm is vastly superior to the alternatives.

Keywords: career prediction; traditional machine learning classifiers; crossfold Validation; boosting methods; XG Boost

\textcopyright 2023 Totem Publisher, Inc. All rights reserved.

1. Introduction

Placement is one of the most important parts of doing well in any graduate or postgraduate course. Every student wants to get hired by top MNCs so they can reach their goals and objectives. Colleges and universities are getting better at finding jobs for their students by giving them better tools and training through training and placement cells. One of the most crucial aspects of a modern education is helping students find solutions where they have failed to get success in their careers. However, there are so many options that it can be hard to choose the right one. Before choosing a career, it's very important to think about how interested you are, how good you are at something, how fast you think it will grow, and how long it will last. Many students have bad grades because they chose a career without thinking about their skills. Choosing the wrong professional path might cost both time and money in the long run. It has also been seen that psychological factors affect choosing the right career path. In particular, students should learn how to understand themselves so they can take part in making decisions about their jobs. But because everyone's goals and ideas are different, it is hard for students to know where they want to go after graduation. From an empirical point of view, on the other hand, you can find out a lot about a student's inner interests and where they plan to go after they graduate by looking at their behavior at school. This makes the behavior of students an important part of their career planning. Modern colleges are increasingly adding sensing, processing, and communication capabilities to their physical facilities as a result of the expansion of information technology. This means that the college information system can trace anything a student does in real time on campus. These kinds of behavioral data can show how each student's habits, skills, preferences, and state of mind are different [1]. Students can also use mining skills to learn more about themselves as they continue to collect this kind of information [2]. ML techniques have been used in recent research to look at the differences and patterns in how different types of college graduates perform [3]. Assumptions can also be used in the real world. For example, we can set up a set of teaching

\textsuperscript{*} Corresponding author.
E-mail address: eswarkoppala@gmail.com
methods based on the real-life situations of college graduates and also make sure that students can get a better education based on their own situations. In this way, students can make career decisions that are relevant to their own life, which helps solve the problem of having difficulty obtaining employment [4]. It is hard to predict a student's career path based on their behavior. Even though the studies that already exist use a variety of machine learning (ML) algorithms, there are problems with low accuracy and models that don't work well. So, based on the theory of social influence [5], we look at the relationship between each student's career prediction and the behavior of students who are similar to him or her. There are a lot of different kinds of data on student behavior, like when they wake up, when they go to sleep, what they do with their friends, what they eat, what sports they play, etc. [6]. Machine learning approaches [7] are used to predict a student's career path based on how they are active. Before we train our model, we put students into groups based on their personal information which affects how we can predict their future careers, get this idea from [8]. In this paper, we specifically look at the behavior of 1020 students who have graduated from college and are now doing their own jobs.

Classification Approaches

In the domain of predicting student careers, classification is the most widely used ML method for modeling and predicting people's behaviors based on their attributes. Correctly assigning class labels to instances with known suitable factors or attribute values but unknown class values is the purpose of classification [9]. Modeling and predicting student actions from a student dataset using ML techniques is challenging since different classifiers produce inconsistent results. As a result, researchers conduct extensive tests with various models based on ML approaches, using student dataset. To evaluate the efficacy of an ML-based prediction model, they employ ten of the most common traditional classification techniques [10], including ZeroR, Naive Bayes (NB), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Adaptive Boosting (AdaBoost), and Logistic Regression (LR) classifiers. These methods are commonly used to foretell a student's future in the workforce, therefore they opted to apply them. There is a brief explanation of the ML classifiers: in the background and related work section. After the ML classifier-based framework has been developed, the performance of each model is evaluated by conducting experiments on real student datasets that contain information about the actions taken by each student in addition to relevant context data. Graduates from a variety of academic disciplines contributed to these datasets. Accuracy, Precision, recall, F-Score, receiver operating characteristic (ROC) value, and error rate are some of the metrics we use to evaluate these classifier-based models in our analysis.

In conclusion, our contributions are as follows:

• At college events, we observe students and record their actions. The consistency of college students' behavior is described using behavioral indices, and the behavioral differences between graduates of alumni and students pursuing their degree are examined. Finally, a feature that takes unlabeled data into account calculates how the data has changed over the years.

• We conduct a comprehensive evaluation on a synthetic dataset comprising over 1,200 alumni to ensure the credibility of our results, we conduct numerous experiments. To develop a model the behavioral information of students belonging to different categories, ML models are proposed. We validate the efficacy of our career prediction method by conducting experiments on a student behavior dataset.

• We used synthetic student datasets of alumni to test each classifier-based prediction model on unseen contextual test cases.

The paper is organized as follows: Section 2 provides an overview of the topic of career prediction for students, as well as an overview of the topic of categorization learning approaches in the literature. In Part 3, we outline the proposed structure and methods. Section 4 provides a description of the synthetic dataset and parameter setup. In Section 5, we discuss the outcomes of our experiments testing the suggested advanced machine learning approach on the student dataset. Section 6 concludes the paper and discusses its potential applications.

2. Literature Survey

Based on the results of an objective test, a computerized career counseling system can provide predictions about the best department for a given individual [11]. The objective of this study is to create a system for career coaching that incorporates a method for predicting a person's compatibility with a certain profession [12]. This work mimics the most popular supervised ML algorithms (DT, RF, KNN, LR, NB, Gradient Boost Tree (GBT), Multi-Linear Perceptron (MLP), and SVM) used to predict academic performance [13]. Using an ML-based hybrid recommendation system, the authors [14] propose a method for personalizing a study plan. The primary goal of this study is to provide a detailed account of the variables utilized by the National Technological University and the implementation of several automated learning approaches in order to get the metrics that allow exhibiting the best algorithm among those evaluated [15]. This study provides a model for
predicting students' academic success based on how they're feeling emotionally, which has the potential to serve as an early warning system, assisting both teachers in monitoring student progress and students in conducting their own assessments of their own performance [16]. LR model, K-NN regression model, DT regression model, XG Boost regression model, GBT regression model, light GBM regression model, and random tree classifier model are all proposed to tackle the student placement prediction problem in this paper [17]. In an effort to improve the institute's training and placement activity, this study aimed to create an automatic system for predicting students' placement at the beginning of their academic careers [18]. In this research, they introduce Harmony, a Deep Learning (DL) driven ML cluster scheduler that optimizes performance by scheduling training jobs to run in isolation from one another (i.e., training completion time). Harmony is founded on a well-thought-out deep reinforcement learning (DRL) framework, enhanced with reward modeling [19]. The experts use ML and Neural Networks to determine if a user meets the criteria for a certain job posting by analyzing their responses to a series of "hyperparameters" [20]. Enhance the quality of educational processes by suggesting a decision support system that provides accurate analysis, improved decision assistance, and reporting and planning capacity to help decision-makers. In this study, we use ML classification approaches to predict the career a graduating engineering student could pursue, with the overarching goal of identifying the elements that influence students' decisions about their futures. The academic performance, athletics, and extracurricular activities of each student are analyzed in order to provide job recommendations using an ML algorithm [21]. In this study [22], they used the machine learning (ML) approach eXtreme Gradient Boosting (XG Boost) to predict students' major selection using a real-world dataset collected at a single university. Precision, recall rate, and F1 value data demonstrate that XG Boost can accurately predict students' job choices at the 89.1%, 85.4%, and 0.872 levels, respectively. C3-IoC (https://c3-ioc.co.uk) is a system introduced in this paper [23] that uses artificial intelligence (AI) to guide students toward appropriate IT career choices based on their individual qualifications. In this study [24], they use two datasets gathered from two Portuguese secondary schools to offer a data mining approach to discover important characteristics and predict student performance. At last, we develop and compare the effectiveness of classification models with SVM, NB, and MLP roots.

Data mining (DM) and ML algorithms were used to describe a model for determining student placement. "Data mining" referred to the process of using ML algorithms to sift through massive datasets in search of relevant information. The authors also made advantage of what they believed to be the superior education data mining technology. It's useful for analyzing whether a student was hired after their campus placement and for predicting how well they did there. Predictions were made using multiple linear regressions, the ML algorithm J48, NB, RF, and Random Tree from the WEKA tool. Higher education organizations can tailor their instruction to the findings [25]. The primary objective of this [26] study was to describe the applications of ML in educational settings, including how institutions can forecast students' performance and what factors should be considered when doing so. The research also evaluates the accuracy of predictions made by various ML systems. By considering a student's learning style, motivation, interest, concentration level, family background, personality type, ability to process knowledge, and testing method, the article concludes that more specific and reliable predictions about how students will do can be made. Class attitude, psychological measures, and student code metrics are used to predict programming class performance [27] [28]. Their qualitative study examines job placement issues of international graduate students going home, migrating abroad, or staying in the US [29]. Their data mining project investigates engineering placement student performance [30], Binary logistic regression predicted student campus placement [31], and Psychology-assisted ML to predict academic success [32]. In the past few years, a lot of work has been done to provide a full review of ML in student career prediction. ML points to possible research areas in the education field. It has specific rules that other fields don't have. ML has been used to predict how well students will do in education life by using different methods and techniques. In Table 1, some of the jobs that different researchers do are listed:

<table>
<thead>
<tr>
<th>S. no.</th>
<th>Algorithms used</th>
<th>Objective</th>
<th>Problem Type</th>
<th>Compared models</th>
<th>Performance metrics</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Voting Ensemble</td>
<td>Student Career Choice</td>
<td>Prediction</td>
<td>NB, K-Tree, SVM</td>
<td>Accuracy, Training Time</td>
<td>[33]</td>
</tr>
<tr>
<td>2</td>
<td>ACCBOX</td>
<td>Career Choice</td>
<td>Prediction</td>
<td>DT, SVM, RF, LR, XG Boost</td>
<td>F1 Score, Micro Precision, Micro Recall</td>
<td>[34]</td>
</tr>
<tr>
<td>3</td>
<td>knn</td>
<td>student career prediction</td>
<td>CF</td>
<td>Accuracy, Error</td>
<td>[35]</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>GB, DL, MLP, RF, LR</td>
<td>student career</td>
<td>Feature Selection</td>
<td>-</td>
<td>AUC, Accuracy, Kappa, RMSE</td>
<td>[36]</td>
</tr>
<tr>
<td>5</td>
<td>XGBoost</td>
<td>career choice</td>
<td>Feature Selection</td>
<td>DT, SVM, RF, GBM</td>
<td>F1 Score, Recall,</td>
<td>[22]</td>
</tr>
<tr>
<td>6</td>
<td>CDMSE</td>
<td>CDD</td>
<td>-</td>
<td>PCA</td>
<td>[37]</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ML Models</td>
<td>Career Prediction</td>
<td>Prediction</td>
<td>DT, RF, SVM, Adaboost</td>
<td>Accuracy</td>
<td>[38]</td>
</tr>
<tr>
<td>8</td>
<td>ML Models</td>
<td>Career Area</td>
<td>Prediction</td>
<td>SVM, RF, DT, XG boost</td>
<td>Accuracy</td>
<td>[39]</td>
</tr>
<tr>
<td>9</td>
<td>ML Models</td>
<td>Student Placement</td>
<td>Prediction</td>
<td>SVM, GNB, KNN, RF, DT, SGD, LR, NN</td>
<td>Precision, Recall, F1-Score, Support, Heatmap</td>
<td>[40]</td>
</tr>
<tr>
<td>10</td>
<td>KNN</td>
<td>Placement Prediction</td>
<td>EDM</td>
<td>RF</td>
<td>Accuracy</td>
<td>[41]</td>
</tr>
</tbody>
</table>
3. Methodology

The problems of career prediction for students cannot yet be fully addressed by the conventional ML approach. By mining the connections between students, we can get a better grasp on the issue of what bonds a group of students together to become alumni. Before discussing how these vast differences between students affect ML approaches, let's examine where they have failed. Luckily, we can get this ahead-of-time data from each alumni. Information bridges allow us to easily link up with similar students. Classification models can be used to model student activity on the basis of synthetic data, which is useful for making predictions about student careers. In the "ML classifiers: literature study" section, we briefly discuss nine of the most common classification techniques that are widely used for predictive purposes and that we use in our analysis. These methods include LR, DT, NB, RF, KNN, SVM, GT Boost, AdaBoost, and Categorical Boosting (CAT Boost). We also consider the proposed eXtreme Gradient Boosting (XGBoost) classifier model in our research as shown in Figure 1, which is gaining traction in the field of machine learning. Here, we outline the XGBOOST method that we've developed.

XG Boost is a popular ML algorithm that excels at a wide range of classification tasks. Thankfully, it excels at solving prediction problems on massive datasets. Here we suggest XG Boost (Algorithm 1) to increase its efficiency in this regard. Like other boosting techniques, XG Boost iteratively constructs an integration model from a classification and regression tree (CART). When constructing trees, each data value starts with the same weight, which is then adjusted based on the results of the analysis. The first round of data values is used to inform the creation of a new classifying model that retains and expands upon the previous round's findings, and so on, until a reliable classifier has been established.

Assume we have a dataset with \( m \) features and \( n \) examples, and that \( D = (x_i, y_i) \) \((|D| = n, x_i \in R^m, y_i \in R)\) where \( D \) is a classification model of a tree as per \( \hat{f}_{i} = \emptyset (x_i) = \sum_{k=1}^{T} f_k(x_i), f_k \in F \) that employs \( K \) additive functions to predict the output.

Algorithm 1. Proposed XG Boost Algorithm for Career Prediction.

1: synthetic student data \( D \)
2: \( D = (x_i, y_i) \) \((|D| = n, x_i \in R^m, y_i \in R)\)
3: \( \hat{f}_{i} = \emptyset (x_i) = \sum_{k=1}^{T} f_k(x_i), f_k \in F \)
   Where \( f_k \) is tree structure
   Define as gap \( F = \{ f(x) = w_q(x) \} \ (q: R^m \rightarrow T, w \in R^T) \)
   where \( T \) is number of leaves
   \( q \) leaf weight
4: Calculate the prediction score of leaves
   \( L(\emptyset) = \sum \hat{y}_i y_i + \sum \delta f_k \) where \( \delta f_k \) is measure the differences
5: Train the model \( U^i = \sum \hat{y}_i y_i^{i-1} + f_i (x_i) + \delta f_k \) where \( f_k \) improves model accuracy
6: Calculate accuracy

![Figure 1. Proposed Research Model](image-url)
4. Dataset Overview and Simulation Setup

In this section, explains how to use different classification techniques to model the student dataset. It has three steps: exploring data sets, processing data and making predictions based on machine learning. Here, we'll talk briefly about each of these steps.

4.1 Prepare Dataset

The dataset that was utilized in this paper was produced from former students who had attended a variety of institutions and graduated with degrees after successfully completing courses in a variety of fields and being placed in a variety of companies, move on to pursue higher education, and do nothing. In order to accomplish this goal, a survey Google form was distributed to all of the alumnae of engineering colleges that are situated in the north coastal region of Andhra Pradesh (India) between April 1, 2019, and December 30, 2022, and responses were gathered from 1027 students after careful communication. This dataset is comprised of a variety of student data kinds, each of which plays an important part in a student's mental ability and physical ability. These ability skills should have an impact on a student's career at any point in time. Overall, the data set that is utilized for the purpose of student placement prediction is an essential component in the process of constructing an accurate and reliable model that has the capacity to accurately forecast the job placement of a student.

4.2 Data Preprocessing

In this study, we modeled individual career prediction in various contexts based on a synthetic dataset compiled from students. To construct the prediction model, we first extract from the dataset the contextual information discussed above. However, the raw contextual data cannot be used with ML techniques to construct the prediction model. To make such student contextual data applicable for building the prediction model, we transform the contextual information into a meaningful category, such as continuous variables into nominal values, fill the missing values, remove duplicates entries, normalize or standardize the numerical features and etc., these datasets use to ML classifiers before building the prediction model.

4.3 Data Set Description

The data for these models come from a dataset of undergraduates. The things that were found to be most important for placing a student were looked at. Table 2 shows the same. The variables used in this study to cover all the parameters that affect student placement. In our dataset, there are 39 attributes which include a particular student's academic information like academic performance (CGPA), attendance, 10th and class XII percentage, technical skills such as different programming languages(C, C++, Java, Python, etc.,) software tools, hardware knowledge and Certifications (Cloud computing, Cyber security, and web development), non-technical skills such as communication skills, teamwork, leadership, creativity, problem-solving ability, logical and analytical thinking ability and their placement information, academic performance data, basic information data of students, family details, behavior data, health reports, regular activates, existing competitive experience, participation in cultural activities, working skills, communication skills, and career prediction data. Student’s behaviors include at home and college and exits, consumption at campus locations, wakeup times, extra cultural activities (e.g., sports, dances), book reading, and academic achievements. Placement Status is the target variable that indicates whether the student was placed in a job or not.

<table>
<thead>
<tr>
<th>Sl No</th>
<th>Attribute Name</th>
<th>Type of attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Name</td>
<td>String</td>
</tr>
<tr>
<td>2</td>
<td>Age</td>
<td>Numeric</td>
</tr>
<tr>
<td>3</td>
<td>Branch</td>
<td>String</td>
</tr>
<tr>
<td>4</td>
<td>Gender</td>
<td>String</td>
</tr>
<tr>
<td>5</td>
<td>Do you have any active backlogs</td>
<td>Boolean</td>
</tr>
<tr>
<td>6</td>
<td>Attendance in %</td>
<td>Numeric</td>
</tr>
<tr>
<td>7</td>
<td>B. Tech CGPA</td>
<td>String</td>
</tr>
<tr>
<td>8</td>
<td>Inter or Diploma Percentage</td>
<td>Numeric</td>
</tr>
<tr>
<td>9</td>
<td>10th Percentage</td>
<td>Numeric</td>
</tr>
<tr>
<td>10</td>
<td>Rate your communication skills</td>
<td>String</td>
</tr>
<tr>
<td>11</td>
<td>Rate your Coding and DBMS skills</td>
<td>String</td>
</tr>
<tr>
<td>12</td>
<td>Rate your Web and Cloud Computing skills</td>
<td>String</td>
</tr>
<tr>
<td>13</td>
<td>Number of certifications you have</td>
<td>Numeric</td>
</tr>
<tr>
<td>14</td>
<td>Have you done any Internships?</td>
<td>Boolean</td>
</tr>
<tr>
<td>15</td>
<td>Is your father a job holder?</td>
<td>Boolean</td>
</tr>
</tbody>
</table>
4.4 Configuring the Simulation Environment

The Simulation Environment, System, and Parameter Setup are all contained within this section. The experiment was carried out on a desktop computer manufactured by ACER (ASPIRE A315-58G). The operating system used was Windows–10 Pro 64-bit, and the processor used was an 11th Generation Intel(R) Core(TM) i5-1135G7 operating system at 2.40 GHz 2.42 GHz. There are 8 GB of RAM and 1 TB of secondary storage. For data analysis, the sk-learn framework and the classification-metrics framework were used. The Pandas framework and the Numpy Python framework were used for data comparison. In Table 3, you can find information about how to set up the parameters. The preprocessed data set can then be split into 70%–30% ratio for training and testing sets, and evaluates the model's performance by hyper parameters are tuned based 10-fold cross validation. In our experiments, we use the accuracy, recall, precision, and micro-F1 as our evaluation metrics.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Config of Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR</td>
<td>{max_iter=1000}</td>
</tr>
<tr>
<td>DT</td>
<td>{random_state=43}</td>
</tr>
<tr>
<td>NB</td>
<td>{random_state=3}</td>
</tr>
<tr>
<td>RF</td>
<td>{n_neighbors=3}</td>
</tr>
<tr>
<td>KNN</td>
<td>{n_neighbors=11}</td>
</tr>
<tr>
<td>SVM</td>
<td>{cache_size=200, tol=0.001}</td>
</tr>
<tr>
<td>ADA BOOST</td>
<td>{base_estimator=dt, classifier_n_estimators=300}</td>
</tr>
<tr>
<td>GBOOST</td>
<td>{learning_rate=0.07}</td>
</tr>
<tr>
<td>C BOOST</td>
<td>{iterations=250, learning_rate=0.01}</td>
</tr>
<tr>
<td>Proposed XGBOOST</td>
<td>{n_estimators = 500, learning_rate = 0.1}</td>
</tr>
</tbody>
</table>

In Figure 2, we see how each dataset is related to all the others via the correlation heatmap. Lighter colors indicate less of a connection between the two sets of data, while blue denotes a close correlation of 1.0 to 0.2. The quality of the synthetic dataset is demonstrated by the increasing correlation of samples over time, with all attributes being highly correlated with other column attributes.

5. Results Analysis

This research enables us to start figuring out the exact factors that affect a student's performance in campus recruitment. It provided an easy-to-understand idea of which considerations directly and indirectly contribute to overall successes and which are the most important things that could help them. We have compared some ML and Advanced ML algorithms such as LR, DT, NB, RF, KNN, SVM, GBoost, CBoost, AdaBoost, and proposed XGBoost. In this research, we have compared the models, by considering parameters like Accuracy, F1 Score, Precision, TPR, FPR and TNR, ROC-AUC values and find the best among them.
In this paper, we have developed ten no ML approaches, that are developed for analyzing student’s placement prediction status. The proposed method Extreme Gradient Boosting has been compared with other nine methods such as LR, DT, NB, RF, KNN, SVM, GBoost[43], CBoost, AdaBoost. Figure 3 Represents the Training and Testing Error analysis of all classifiers along with Extreme Gradient Boosting method. It is obvious to conclude that, the proposed method is having less error rate in terms of both training and testing as compared to other considered approaches. Table 4 indicates the result analysis of all the methods in terms of several performance indicators such as accuracy, TP, FP, TN, F1 Score and ROC-AUC. Out of nine compared methods, the accuracy of XGBoost (86%) is high. It has more number of TP (101), TN(100) values and less number of FP(18), FN(14) values. Similarly, the TPR of the proposed method is 0.87, and TNR is 0.84 which are highest among the remaining. The FPR of the proposed method is 0.15 which least among the all. By considering all the above results, the performance of the proposed Extreme Gradient Tree Boosting method is quite encouraging for effective prediction of student placement status.
Figure 3. Training and Testing Error analysis of all classifiers

Table 4. Results of all the methods along with Proposed XGBoost

<table>
<thead>
<tr>
<th>Research Models</th>
<th>Metrics for Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td>LR</td>
<td>0.78</td>
</tr>
<tr>
<td>DT</td>
<td>0.72</td>
</tr>
<tr>
<td>NB</td>
<td>0.77</td>
</tr>
<tr>
<td>RF</td>
<td>0.72</td>
</tr>
<tr>
<td>KNN</td>
<td>0.82</td>
</tr>
<tr>
<td>SVM</td>
<td>0.76</td>
</tr>
<tr>
<td>ADA BOOST</td>
<td>0.76</td>
</tr>
<tr>
<td>GBOOST</td>
<td>0.79</td>
</tr>
<tr>
<td>CAJ BOOST</td>
<td>0.75</td>
</tr>
<tr>
<td>Proposed XGBOOST</td>
<td>0.86</td>
</tr>
</tbody>
</table>

Figure 4 represents the ROC curves of all the methods i.e. a)LR, b) DT c) NB, d) RF e) KNN f) SVM g) GBOOST h) ADA BOOST i) CBOOST and j) PROPOSED XGBOOST. Here the ROC-AUC values of every algorithm are greater than 0.5, which justifies that every algorithm has high TP Rate and less FP Rate. The training error and testing error of the proposed method are 0.00 and 0.14 respectively (Shown in Table 5). By considering all the required parameters, we came with a conclusion that Extreme Gradient Tree Boosting is giving the best results for our dataset.
The experimental results show that our method is superior to its state-of-the-art alternatives, demonstrating the positive impact that the XG Boost technique has on the overall performance of the model.

6. Conclusion

In this research, we examine how professional competencies, behavioral consistency, and other factors influence the career paths of college students. The research has also provided numerous useful suggestions for enhancing the model. This research aims to bridge the gap between traditional machine learning concepts and advanced machine learning algorithms and the experimental categorization of students for career prediction. We have proposed an XG Boost method that incorporates each institution's a priori knowledge. Assumptions that items in the same category should all share the same label inspired this. Several trials have shown that compared to other methods of career prediction, ours produces significantly more accurate outcomes. There are interesting avenues for research to take in the future. There is a more precise way to discover initial classification models. Our approach can also be expanded to incorporate additional types of data, such as academic performance and survey responses, in addition to behavioral observations. Additionally, it makes sense to enhance our model to not only anticipate job choices but also advise on career planning, such as recommending the necessary courses.


Enhancing Deception Detection with Exclusive Visual Features using Deep Learning

Victor Diaz\textsuperscript{a}, W. Eric Wong\textsuperscript{b,*}, and Zizhao Chen\textsuperscript{b}

\textsuperscript{a}California Polytechnic State University, Humboldt, USA
\textsuperscript{b}University of Texas at Dallas, Richardson, Texas, USA

Abstract

A combination of nonverbal cues, verbal cues, and measurements of body abnormality make guidelines to determine deceitfulness. The combination of these guidelines will vary from person to person, making deception detection a complex challenge. Research has demonstrated that the accuracy of the latest computerized polygraph testing techniques is 98\% accurate. Several human-controlled variables help to achieve this level of accuracy, such as being properly trained and must use an accepted procedure and scoring system from the British Polygraph Society. This causes a lack of availability for Deception detection as the implementing these techniques have training from the British Polygraph Society. Hence this research aims to reduce the requirements of lie detection by relying on Visual Features tracked with computer vision. The proposed multi-modal will track facial and body movements to classify whether a person is Deceiving or telling the Truth. The model proposed will use data consisting of videos collected from public court trials. The data will be cleaned with Facial Action Units (AU) with OpenFace, and then augmented with various rotations. The features extracted from the videos are the Movement with Holistic landmarks and Unique features from deep learning extraction. The Multi-model will consist of three pathways: a 3D-CNN pathway, a CovLSTM2D Pathway, and a dense pathway. The outputs of the three paths are concatenated and fed into a dense layer with SoftMax activation for classification. With a continuous emphasis on examining the proposed methodology for model creation, we discovered that higher accuracy can be achieved by leveraging deep learning algorithms for visual inputs as complex as the human body.

Keywords: deception detecting; exclusive visual; computer vision

© 2023 Totem Publisher, Inc. All rights reserved.

1. Introduction

Computer vision constitutes a specialized field within the realm of artificial intelligence, concentrating on the extraction of pertinent information from diverse forms of visual input such as images and videos. In stark contrast to the innate ability of humans to perceive objects and ascribe significance to them, computers are inherently devoid of this capability. However, through the application of machine learning techniques, computers can be empowered to emulate the cognitive processes of humans [1]. Within the domain of computer vision, the integration of these machine learning algorithms serves as a conduit for the classification, extraction, and discernment of data, leading to the proficient identification and labeling of objects [2].

Machine learning has become an indispensable tool in the realm of computer vision, facilitating remarkable advancements in various applications. This synergy between machine learning and computer vision is exemplified by the utilization of convolutional neural networks (CNNs), a class of deep learning models, which have demonstrated exceptional performance in tasks such as image classification and object detection [3, 4]. The ability of CNNs to automatically extract hierarchical features from raw pixel data has revolutionized image analysis, allowing systems to recognize patterns and objects with remarkable accuracy. Furthermore, the application of recurrent neural networks (RNNs) in tasks like image captioning has enabled machines to describe visual content with human-like fluency [5]. These machine learning techniques are instrumental in enhancing the capabilities of computer vision systems, enabling them to decipher complex visual information and opening doors to a wide array of applications, from autonomous vehicles to medical image analysis.

* Corresponding author.
E-mail address: ewong@utdallas.edu
Behavior/Emotions are one of the interests of computer vision. There are many challenges when annotating emotions with machine learning [6]. According to the American Psychological Association: “Emotions are conscious mental reactions (such as anger or fear) subjectively experienced as strong feelings usually directed toward a specific object and typically accompanied by physiological and behavioral changes in the body. [7]” The state of mind of an individual affects the data used in machine learning training, which makes notating emotions with machine learning challenging [8].

In a recent study, computerized polygraph testing techniques have a 98% average accuracy in field examinations, 92% in field-independent analysis, 80% in laboratory simulations, and 81% in laboratory-independent analysis [2]. While the accuracy level makes deception detection a viable tool, the methodology limits the availability of such techniques. The implementation of the methodology is limited to examiners who undergo continuous professional development and have qualified certification. The methodology consists of using the latest techniques and methods in the field, being a current member of the British Polygraph Society or the American Polygraph Association, and submitting work for quality checks as part of a commitment to maintaining the highest testing standards possible. In addition, having the minimum recording equipment is needed for cardiovascular, respiratory, and electrodermal activity [8]. With computer vision, the goal is to reduce the input requirement to only video footage.

To narrow the scope of this research, we focus on safety-critical environments. Safety-critical environments are scenarios where deception or intentionally misleading someone can cause safety concerns. In court, misinformation can make the availability of computer vision in critical environments such as courts and trials. In a critical environment where a person’s future is at stake, having a tool that can help a person’s decisions and is accessible at any time is valuable. Hence this research aims to increase the availability of deception detection with computer vision by proposing a multimodal which only uses Visual feature extraction.

The paper is structured as follows: Section 2 provides a comprehensive overview of computer vision fundamentals, including an assessment of previously proposed models and related research. Section 3 outlines the research methodology and implementation specifics, while Section 4 details the experimental procedures, including the creation of both the primary and sub-models. Section 5 focuses on presenting and analyzing performance results during validation and testing. Finally, Section 6 offers conclusions drawn from the research and identifies future directions for further investigation, ensuring a logical and coherent flow of information throughout the paper.

2. Related Work

2.1 Visual Features Detection

Studies on visual cognition show that humans do not simultaneously concentrate on an entire scene. Instead, they sequentially direct their attention to various portions of the scene to extract pertinent information [11]. Action recognition concerns finding meaning from a set temporal visual field information. Similar to how the brain works. Visual features are the image frame-based information derived from videos. In Deception detection, we use different features extracted from visual data, such as Acoustical features (Voice or audio), textual features (text encoding or transcripts), or Visual features (image frame-based information). The filtering stage is a standard start with biological models [12]. And depending on what feature filter is applied, that data can improve the Machine learning dome [13]. From the Visual input of frame data in Figure 1 we can see an example of information extraction. Now know that the frame contains facial features.

![Figure 1. Example of information being extracted from a frame](image)
Numerous publications on deception detection have undertaken the extraction of visual features and subsequent classification. These studies have included experiments as Table 1:

The Face Landmark Detection System and Azure Machine Learning utilized two algorithms: the Two-Class Support Vector Machine and Linear Regression. The control method involving human subjects achieved an accuracy of 76.2% [14]. Facial displays and Non-Verbal features achieved an accuracy of 68.59% through Decision Tree classification and 73.55% through Random Forest tree classification [15].

The support Vector model achieved an accuracy of 67.20% when utilizing Action Units extracted from OpenFace. To ensure the multimodal model's validity and performance, various models and classifiers were tested against it. The model being tested are a subset of the proposed model. Including single modality video classifying with 3D-CNN, Convolutional LSTM 2D, and Holistic landmark Random Forest Classification. Multi-modal model video classification with the combination of 3D-CNN and ConvLSTM-2D. Multi-modal model video classification with the concatenation of 3D-CNN and ConvLSTM-2D and Random Forest Classification. To ensure performed enhancements at each step, the test set being compared is constant.

By employing PittPatt to locate the facial region and an Anthropometric face model, we achieved an accuracy of 76.92% when analyzing macro expressions. Additionally, they obtained an accuracy of 56.92% when examining micro-expressions using a Random Forest model [16]. Visual Features extracted visual features from the videos using a 3D CNN and employing a 3D CNN classifier; we achieved an accuracy of 78.57% [17]. Utilizing a CNN for extracting visual features, combining IDT with a Linear Support Vector Machine achieved an accuracy of 77.31%. Furthermore, using Micro-expression with a Random Forest model resulted in an accuracy of 80.64%. Using a linear regression model, the integration of both Micro-expressions and IDT reached an accuracy of 89.88% [18]. Visual features extracted from 3D-CNN, which achieved an accuracy of 95.96% when utilized in a Multi-Layer Perceptron model [19]. Visual Features, specifically face landmark detection and head pose estimation using Convolutional Experts Constrained Local Model (CE-CLM), achieved the following accuracies with different classification models: Logistic Regression - 54.86%, Random Forest - 58.42%, SVM (Linear) - 65.56%, SVM (Sigmoid) - 71.99%, SVM (RBF) - 76.84% [20].

Visual features extracted from OpenFace and applied Long Short-Term Memory (LSTM), which achieved an accuracy of 56%. Additionally, utilizing a Support Vector Machine (SVM), an accuracy of 57.4% was obtained [21].

Visual features are extracted from an R-CNN network, enabling the creation of a Face-focused cross-stream network (FFCS). The FFCS incorporates fusion with two inputs: focusing on the face and movement within the network. By incorporating adversarial learning, meta-learning, and cross-stream correlation learning into the model, they have achieved an accuracy of 93.16%. Additionally, the model reached a face detection accuracy of 84.33% [22].

<table>
<thead>
<tr>
<th>Method/Model Description</th>
<th>Accuracy</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two-Class Support Vector Machine and Linear Regression</td>
<td>76.20%</td>
<td>[14]</td>
</tr>
<tr>
<td>Decision Tree classification (Facial displays)</td>
<td>68.59%</td>
<td>[15]</td>
</tr>
<tr>
<td>Random Forest tree classification (Facial displays)</td>
<td>73.55%</td>
<td>[15]</td>
</tr>
<tr>
<td>PittPatt and Anthropometric face model (Macro expressions)</td>
<td>76.92%</td>
<td>[16]</td>
</tr>
<tr>
<td>Random Forest model (Micro-expressions)</td>
<td>56.92%</td>
<td>[16]</td>
</tr>
<tr>
<td>Visual Features with 3D CNN</td>
<td>78.57%</td>
<td>[17]</td>
</tr>
<tr>
<td>Integration of Micro-expressions and IDT (Linear Regression)</td>
<td>89.88%</td>
<td>[18]</td>
</tr>
<tr>
<td>Visual Features from 3D-CNN (Multi-Layer Perceptron)</td>
<td>95.96%</td>
<td>[19]</td>
</tr>
<tr>
<td>Visual Features (CE-CLM) - Logistic Regression</td>
<td>54.86%</td>
<td>[20]</td>
</tr>
<tr>
<td>Visual Features (CE-CLM) - Random Forest</td>
<td>58.42%</td>
<td>[20]</td>
</tr>
<tr>
<td>Visual Features (CE-CLM) - SVM (Linear)</td>
<td>65.56%</td>
<td>[20]</td>
</tr>
<tr>
<td>Visual Features (CE-CLM) - SVM (Sigmoid)</td>
<td>71.99%</td>
<td>[20]</td>
</tr>
<tr>
<td>Visual Features (CE-CLM) - SVM (RBF)</td>
<td>76.84%</td>
<td>[20]</td>
</tr>
<tr>
<td>Visual Features from OpenFace - LSTM</td>
<td>56%</td>
<td>[21]</td>
</tr>
<tr>
<td>Visual Features from OpenFace - SVM</td>
<td>57.40%</td>
<td>[21]</td>
</tr>
<tr>
<td>Face-focused cross-stream network (FFCS)</td>
<td>93.16%</td>
<td>[22]</td>
</tr>
<tr>
<td>Face detection accuracy (FFCS)</td>
<td>84.33%</td>
<td>[22]</td>
</tr>
</tbody>
</table>
2.2 Multimodal Features Deception Detection

The multimodal approach uses a combination of different features to train the model:

Verbal Features consisting of unigrams and bigrams, along with Non-Verbal Features involving Facial and Gesture display, an accuracy of 75.20% reached by Decision Tree classification and 50.41% with Random Forest Tree classification [15].

A combination of Lexical Feature Extraction, Audio-based Features, and Visual Features reached an accuracy of 78.95% with Feature-level Fusion, 76.12% with Decision-level Fusion, and 74.02% with Utterance-based Feature Fusion in a Vector Machine model [23].

A combination of Macro and Micro-expression reached an accuracy of 76.92% with Random Forest classification [16].

Audio, visual, and textual features extracted from a Convolutional Neural Network (CNN) and openSMILE were combined using early and late feature fusion techniques. The result was an accuracy of 92% with early fusion and 96.4% when using late fusion [17].

A combination of Acoustic, Gestures, Text Modality, and IDT (Improved Dense Trajectory) features achieved the following accuracies with different classifiers: 87.73% with L-SVM (Linear Supported Vector Machine), 82.33% with K-SVM (K-supported vector Machine), 77.76% with NB (Naive Bayes), 77.77% with DT (Decision Tree), 84.77% with RF (Random Forest Tree), 78.94% with LR (Linear Regression), and 78.99% with AdaBoost [18].

A combination of feature extraction by Acoustic, Gestures, and Text Modalities was used. Gesture feature extraction involved the utilization of a 3D-CNN to identify facial expressions. Textual feature extraction employed a Convolutional Neural Network (CNN). For the audio feature, openSMILE was utilized. The facial expressions were manually annotated into binary features. The algorithm achieved an accuracy of 75.20% for DT (Decision Tree), 50.41% for RF (Random Forest), 90.49% for MLPC (Multi-Layer Perceptron Classifier), and 90.99% for MLPH+C (Multi-Layer Perceptron with Hidden Layer Count) [19].

A combination of Visual, Acoustical, and Textual modalities was employed. OpenFace was utilized for Visual modality extraction, FFmpeg was used to extract the Acoustics modality, and Watson's Speech to Text was employed for textual modality extraction. Through early and late fusion techniques, an accuracy of 0.665 was achieved using Deep Learning, and an accuracy of 0.610 was achieved using LSTM [15].

3. Methodology

3.1 Machine Learning Models

The Multi-Modal model structure (as shown in Figure 2) is a multi-model that uses deep-learning feature extraction to detect deception. Deception Detection requires an architecture that implements sequencing or the Temporal Dimension. Events such as looking away or stuttering are not always cues of deception [24]. While deception in itself does not affect someone's behavior, deceptive indicators are signs of attempting behavior control. This implies that deceptive behavior may be visible if a liar experience telling the truth. There is no generalized deceptive behavior for everyone; some behaviors are more likely to occur than others [25]. Knowing there can be some nonverbal information from deception behavior, the proposed ML model will detect Truthful and deceptive behavior.

The proposed model must be attuned to frame sequences/video and can aptly identify a person. Furthermore, it is why 3D-CNN, 2D-LSTM, and Holistic landmark features are used. 3D-CNN and 2D-LSTM have shown generalized success in representing temporal data [26, 27] Holistic landmarks accurately represent people with set landmarks [28]. Random Forest classifiers determine the relative importance by considering the collective predictions of all constituent decision trees [29]. Since nonverbal deception detection does not generalize and is situational, Random Forest classification is used for the final classification.

To ensure the multimodal model's validity and performance, various models and classifiers were tested against it. The model being tested are a subset of the proposed model. Including single modality video classifying with 3D-CNN, Convolutional LSTM 2D, and Holistic landmark Random Forest Classification. Multi-modal model video classification with the combination of 3D-CNN and ConvLSTM-2D. Multi-modal model video classification with the concatenation of 3D-CNN and ConvLSTM-2D and Random Forest Classification. To ensure performed enhancements at each step, the test set being compared is constant.
3.2 Feature extraction

Studies have shown that there exist some indications of deception [30, 31]. The issues with non-verbal deception detection stem from humans' cognitive complexities and the inability to have every characteristic in a controlled environment. [31, 32].

Three significant gaps have been identified when dealing with Non-verbal Deception research; increase the methodology domain by examination, increase the diagnostic of cues for deception work, and heighten focus where there is no alternative to making nonverbal veracity assessments [3].

With AI, we address the first concerns by creating and testing a variety of ML methodologies against each other; the second concern and third concerns by using deep learning feature extraction, which removes the issues of previous diagnosis cues and lets the model create its own identify features from the given data.

Feature extraction allows a machine learning model to consume identity characteristics for classification. This technique yields better results when compared to feeding raw data to an ML model. [33, 34]. This model uses two Deep learning feature extraction methods, a Conv LSTM 2d and 3D CNN and a Holistic Landmark extraction. The use of automatic feature constructions from Deep learning addresses the issues of finding identified characteristics [33, 34]. We can identify features from the data given with these deep learning models [34].

The last feature used in the model is MediaPipe Holistic Landmarks which combines components of the pose, face, and hand landmarks to create a complete landmark for the human body. With 3DCnn and CovLSTM2d, we input face frames to get a complete view of the human body Holistic Landmarks were added.

3.3 Paths for Extraction

A Convolutional LSTM 2D (CovLSTM2d) is a deep learning model. It Combines the convolutional structures (CNN) with the feedback connection of the LSTM [33]. Combining the input sequence data from a CNN makes it well-suited for image and video Classification [35]. CovLSTM2d models explicitly model temporal dependencies using LSTM cells [33]. The Convolutional LSTM, as shown in Figure 3 is a network defined with three different layers. The first layer has 4 filters and uses a 3x3 kernel size with a hyperbolic tangent (tanh) activation function. It employs the 'channels_last' data format and includes a recurrent dropout of 0.2 while returning sequences. After that, a Batch Normalization layer is applied. Following that, a MaxPooling3D layer is used with a pool size of (1,2,2) and 'same' padding in the 'channels_last' data format.
The same structure is repeated two more times but with different filter numbers (8 and 16) for the ConvLSTM2D layers. After each ConvLSTM2D layer, Batch Normalization and MaxPooling3D layers are used again, along with a Time Distributed layer and Dropout layer with a dropout rate of 0.2 applied to each time step.

Finally, a flattened layer is employed to transform the output feature map into a one-dimensional vector, which can be used for further processing or classification tasks.

Figure 3. Diagram of Convolutional LSTM 2D path

MediaPipe Holistic Landmarks combines components of the viewing angle of the purpose, face, and hand landmarks to create a complete landmark for the human body.

In this process for Holistic Landmark extraction, the color space is converted to RGB using the function Convert_ColorSpaceToRGB(). Then, the pose landmarks are obtained using the Results.pose_landmarks() method, and the coordinates of each landmark are collected and appended to a list using the append(coordinates_of_each_landmark) operation. After gathering the individual landmark coordinates, they are aggregated to extract additional features using the Aggreatege_feateurs_appended() function. Finally, the resulting output features are obtained and ready for further analysis or usage.

A 3D CNN as shown in Figure 4 uses a three-dimensional filter to perform convolutions. It learns to recognize patterns in videos from Height, Width, and Depth. The depth in this context will be the temporal dimension. 3DCNN treats the temporal dimension in the same way, as the spatial dimension(time)[34, 36]. A 3D convolutional neural network (Conv3D) is described with three layers. The first Conv3D layer has 32 filters and uses a 3x3x3 kernel size with the Rectified Linear Unit (ReLU) activation function. It employs ‘same’ padding to preserve the spatial dimensions. After that, a MaxPooling3D layer is used with a pool size of (1, 2, 2) to reduce the spatial dimensions while keeping the depth unchanged.

The same structure is repeated two more times, but with different filter numbers (64 and 128) for the Conv3D layers. Each Conv3D layer is followed by a MaxPooling3D layer with the same pooling size.

After the last MaxPooling3D layer, a flattened layer is applied to transform the output feature map into a one-dimensional vector. This vector is then fed into a Dense layer with 1024 units and ReLU activation to perform fully connected processing. The final output represents the extracted features that can be utilized for various tasks like classification or regression.

Figure 4. Diagram of 3D CNN path
4. Experiment

4.1 Experiment Setup

The core dataset utilized for detecting deceitful behavior comprises real-life trail-based videos, which have been employed in numerous studies [15]. The database is composed of 121 trial videos split between 61 Deceptive and 60 Truthful. Video labeling was done manually primarily based on the court verdicts, posterior exenteration, verification of police reports against declarations, and other relevant factors [15].

The data was divided as seen in Table 2 into sections to prevent the testing data from mixing with the training data and creating Train-Test Contamination.

<table>
<thead>
<tr>
<th>Table 2. Division of unedited video files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Truth</td>
</tr>
<tr>
<td>Training</td>
</tr>
<tr>
<td>Testing</td>
</tr>
</tbody>
</table>

There are prominent issues with training data in machine learning, such as Insufficient Data, Non-Representative Training data, Poor Quality Data, or Underfitting the Data [34]. Input diversity and quantity of the dataset are essential for machine learning [24].

4.2 Data augmentation

Data augmentation and cleaning as seen in Figure 5 will be used on the data set to increase our training set. The data cleaning will work in two parts: The first step is to divide all the videos into videos of 64 frames with OpenCV, which creates 97 inputs into a data set of 1095.

![Data Augmentation Diagram](image)

The second step is taking those 1095 videos and capturing all the faces from them using OpenCV. This reduces our data set to 1027 as some faces can not be identified in every video. The full division of the Test/Train set is seen in Table 3.

<table>
<thead>
<tr>
<th>Table 3. Division of augmented video files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instance</td>
</tr>
<tr>
<td>Training</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Testing</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
To ensure the enhanced performance and validity of the multimodal model, each step and classifier within the model was rigorously tested. Firstly, the 3D-CNN deception detection step involved training a 3D-CNN model on face-cropped videos consisting of 64 frames. A total of 20 tests were conducted, with 10 tests performed on the testing set containing 64-frame videos and the remaining 10 on the original video set.

Next, the ConvLSTM-2D deception detection step utilized face-cropped videos with 64 frames to train a ConvLSTM-2D model. Three tests were conducted: the first test involved a 64-frame test set, the second utilized the original video set, and the third involved a training split extracted from the original training set.

For holistic landmarks with random forest classification, face-cropped videos with 64 frames were used to extract holistic landmark features. A random forest classifier with 100 trees was trained using these extracted features. Similarly, three tests were conducted: the first with a 64-frame test set, the second with the original video set, and the third with a training split from the original training set.

Overall, each component of the multimodal model was meticulously tested, and the evaluation was performed under different conditions to ensure its effectiveness and generalization across various datasets and scenarios.

4.4 Multi-Modal model

Each step and classifier were tested to ensure the validity and increased performance of the multi-modal model. The proposed approach involves a Multi-modal model that fuses two distinct pathways, namely, a 3D Convolutional Neural Network (3D-CNN) and a Convolutional LSTM-2D model. The input to this model consists of face-cropped videos containing 64 frames, which are utilized during the training process. The model comprises two parallel paths: Path 1 represents the 3D-CNN model, and Path 2 represents the ConvLSTM model. These two paths operate independently on the input data.

To create a unified representation, the outputs from both Path 1 and Path 2 are concatenated and fed into a dense layer. This dense layer further transforms the aggregated information and produces initial outputs, which are then passed through a softmax function to obtain the final predictions.

To evaluate the performance of the proposed Multi-modal model, three different tests were conducted. The first test used a test set containing 64 frames per video. The second test utilized the original video set as the input. Lastly, the third test involved a training split taken from the original training set. These tests were carried out to assess the model's effectiveness and generalization across various scenarios and datasets.

A multi-modal model was developed, which integrates feature extraction from both a 3D-CNN and a ConvLSTM-2D model, and the classification is performed using a Random Forest classifier. The input to the model consists of face-cropped videos containing 64 frames, which are utilized in a two-path Multi-Modal model. Path 1 represents the 3D-CNN model, while Path 2 represents the ConvLSTM model. These two paths operate independently on the input data.

To create a unified representation of the features extracted by both paths, the outputs from Paths 1 and 2 are concatenated and fed into a dense layer. This dense layer further processes the combined features, and the initial outputs are obtained using a softmax function.

Subsequently, the features extracted from the multi-modal model are used for classification. These features are fitted to a Random Forest classifier comprising 100 trees. The Random Forest classifier allows for effective classification based on the combined information obtained from both the 3D-CNN and ConvLSTM-2D models.

To assess the performance and generalization of the multi-modal model, three different tests were conducted. The first test involved using a test set with 64 frames per video. The second test utilized the original video set as the input. Lastly, the third test involved using a training split extracted from the original training set. These tests were crucial in evaluating the model's capability to detect and classify deception across various datasets and scenarios.

The proposed multi-modal model combines feature extraction from three distinct sources: a 3D-CNN model, a ConvLSTM-2D model, and Holistic landmarks. These features are then integrated and classified using a Random Forest classifier. The model is trained on face-cropped videos containing 64 frames, and it operates within a tree-path Multi-Modal framework.
Path 1 represents the 3D-CNN model, while Path 2 represents the ConvLSTM model. These paths process the input data independently, extracting relevant features from each frame of the video. Additionally, Path 3 involves Holistic landmark feature extraction from the face-cropped video, where the extracted landmarks are aggregated to form meaningful features.

The outputs from Path 1 and Path 2 are concatenated and passed through a dense layer with an initial softmax function, resulting in a unified representation of the extracted features from both 3D-CNN and ConvLSTM-2D models.

To incorporate the Holistic Landmark features, all the extracted features, including those from the 3D-CNN, ConvLSTM-2D, and Holistic landmarks, are concatenated to form a single feature set for each video.

Subsequently, the combined features from all three paths are utilized as input to a random forest classifier consisting of 100 trees. This classifier leverages integrated information to perform deception detection with enhanced accuracy and robustness.

To evaluate the performance and generalization of the multi-modal model, three different tests were conducted. The first test involved using a test set with 64 frames per video. The second test utilized the original video set as the input. Lastly, the third test involved using a training split extracted from the original training set. These tests were crucial in assessing the model’s effectiveness in detecting deception across various datasets and scenarios.

### 5. Results

As shown in Table 4, after completing the experiment, the first subset single model, a 3D-CNN classifier trained with face augmentation, achieved an accuracy of 0.4166667 when tested against raw unprocessed test files. When applied to the processed test set consisting of 64 frames, it reached an accuracy of 0.45173745. Furthermore, when tested with the split training set of videos, it achieved an accuracy of 0.4854369.

The second subset single model, a ConvLSTM2D classifier also trained with face augmentation, achieved an accuracy of 0.4166666567 with raw unprocessed test files. When tested with the processed set of 64 frames, it achieved an accuracy of 0.509652495. Furthermore, testing with the split training set of videos yielded an accuracy of 0.91176470588.

The third subset single model employed Holistic landmarks with a Random Forest classifier. Trained with face augmentation, this model achieved an accuracy of 0.38 when tested against raw unprocessed test files. The accuracy rose to 0.39 when the model was applied to the processed set of 64 frames. Additionally, when tested with the split training set of videos, the model reached an accuracy of 0.79.

The first subset multi-modal model, which combines 3D-CNN and ConvLSTM-2D, was trained with face augmentation and achieved an accuracy of 0.625 when tested against raw unprocessed test files. When the model was applied to the processed test set of 64 frames, it reached an accuracy of 0.4438502673. Furthermore, testing with the split training set of videos led to an accuracy of 0.4438502673.

The second subset multi-modal model combined feature extraction from 3D-CNN and ConvLSTM-2D and was classified with a Random Forest Classifier. Trained with face augmentation, it reached an accuracy of 0.54166667 when tested against raw unprocessed test files. When applied to the processed test set of 64 frames, it achieved an accuracy of 0.490347490. Furthermore, when tested with the split training set of videos, it reached an accuracy of 0.74866632.
The proposed multi-modal model integrates three distinct feature extraction methods: 3D-CNN, ConvLSTM-2D, and Holistic landmarks, employing a Random Forest Classifier for classification. Its performance surpasses that of individual models such as CNN or ConvLSTM-2D. The initial accuracy of 0.54 achieved on raw test data indicates the model's proficiency in pattern recognition. A more noteworthy performance emerges when the model is applied to processed data containing 64 frames, yielding an accuracy of 0.56. This indicates an enhanced capacity to comprehend intricate temporal sequences. Additionally, the model demonstrates an accuracy of 0.87 when tested on a segregated training video dataset, highlighting its robust generalization capabilities to unseen data instances. In contrast to standalone CNN or ConvLSTM-2D models, the multi-modal model's merit lies in its amalgamation of techniques, harnessing their respective advantages while mitigating limitations. This amalgamation facilitates the acquisition of both spatial and temporal features, resulting in improved predictive abilities.

6. Conclusion

This study aimed to create a machine-learning model proficient at detecting deception solely through video inputs. We designed an experimental model that leverages 3D-CNN, ConvLSTM-2D, and MediaPipe Holistic Landmark feature extraction methods, followed by classification through Random Forests. We used a dataset of real-life, trial-based videos to train and test our model.

Based on the results in Table 4, single models like 3D-CNN and ConvLSTM2D achieved modest accuracy ranging from 0.42 to 0.91. However, the real breakthrough came with the multi-modal model, integrating 3D-CNN, ConvLSTM-2D, and Holistic landmarks with a Random Forest Classifier. It demonstrated remarkable accuracy, reaching 0.87 on a separate training video dataset, highlighting its ability to generalize effectively. This underscores the potential of multi-modal approaches, surpassing the limitations of individual models like CNN or ConvLSTM-2D and significantly improving predictive performance.

Our findings suggest that while the model performed with lower accuracy on raw test files, it showed promise in the face of inherent challenges. Despite these challenges, the model using Deep learning did surpass the 80% accuracy as shown in Figure 6, indicating that our study's high-level structure with deep learning Emphasis can yield promising results.

![Figure 6. Comparison of accuracy using different models](image)

7. Future Work

However, it is crucial to recognize that certain limitations in our approach could enhance our findings' overall accuracy and generalizability. One such limitation lies in the absence of an exhaustive parameter-tuning process and data limitation. The unpredictable factors in raw test files present a unique obstacle, yet our proposed model exhibited a general accuracy trend in these conditions. Machine learning models typically perform better with larger, more diverse datasets - a factor that might have yet to be fully realized with our current dataset. Parameter tuning is a critical aspect of machine learning models that enhance performance by optimizing their internal parameters or hyperparameters. Hyperparameters are the model elements we adjust before the training begins, such as the learning rate, regularization parameters, the depth of decision trees, etc.

Our results show that the availability of deception detection could increase with the work of machine learning and AI. Nonetheless, it is essential to remember that deception detection is a complex and multi-faceted task. Therefore, the predictions made by a machine learning model should not be the sole determinants of truthfulness but should be used as supplementary tools within a broader context.
8. Acknowledgment

This work was supported in part by the U.S. National Science Foundation under Grant 1757828, Grant 1822137, and Grant 2050869. This work was completed solely by the author at the University of Texas at Dallas Research Experience for Undergraduates (REU) under the supervision of Dr. W. Eric Wong and Mr. Zizhao Chen.

References


