

# **APPLIED COMPUTER SCIENCE**

The Journal is a peer-reviewed, international, multidisciplinary journal covering a broad spectrum of topics of computer application in production engineering, technology, management and economy.

The main purpose of Applied Computer Science is to publish the results of cutting-edge research advancing the concepts, theories and implementation of novel solutions in computer technology. Papers presenting original research results related to applications of computer technology in production engineering, management, economy and technology are welcomed.

We welcome original papers written in English. The Journal also publishes technical briefs, discussions of previously published papers, book reviews, and editorials. Especially we welcome papers which deals with the problem of computer applications in such areas as:

- manufacturing,
- engineering,
- technology,
- designing,
- organization,
- management,
- economics,
- innovations,
- competitiveness,
- quality and costs.

The Journal is published quarterly and is indexed in: BazTech, Cabell's Directory, Central & Eastern European Academic Source (CEEAS), CNKI Scholar (China National Knowledge Infrastructure), DOAJ (Directory of Open Access Journals), EBSCO, ERIH PLUS, Index Copernicus, J-Gate, Google Scholar, Scope Database, Scopus, TEMA Technik und Management.

Letters to the Editor-in-Chief or Editorial Secretary are highly encouraged.



## CONTENTS

Rafał KLIZA, Karol ŚCISŁOWSKI, Ksenia SIADKOWSKA, Jacek PADYJASEK, Mirosław WENDEKER <b>STRENGTH ANALYSIS OF A PROTOTYPE COMPOSITE HELICOPTER ROTOR BLADE SPAR.....</b>	<b>5</b>
Kevin Joy DSOUZA, Zahid Ahmed ANSARI <b>HISTOPATHOLOGY IMAGE CLASSIFICATION USING HYBRID PARALLEL STRUCTURED DEEP-CNN MODELS.....</b>	<b>20</b>
Jakub ANCZARSKI, Adrian BOCHEN, Marcin GŁĄB, Mikołaj JACHOWICZ, Jacek CABAN, Radosław CECHOWICZ <b>A METHOD OF VERIFYING THE ROBOT'S TRAJECTORY FOR GOALS WITH A SHARED WORKSPACE.....</b>	<b>37</b>
Anusha NALLAPAREDDY, Sneha BANAVATH, Vihassitha GUNDAM <b>DETECTION AND CLASSIFICATION OF VEGETATION AREAS FROM RED AND NEAR INFRARED BANDS OF LANDSAT-8 OPTICAL SATELLITE IMAGE.....</b>	<b>45</b>
Kuba ROSŁANIEC <b>ANALYSIS OF THE EFFECT OF PROJECTILE IMPACT ANGLE ON THE PUNCTURE OF A STEEL PLATE USING THE FINITE ELEMENT METHOD IN ABAQUS SOFTWARE.....</b>	<b>56</b>
Lubna RIYAZ, Muheet Ahmed BUTT, Majid ZAMAN <b>IMPROVING CORONARY HEART DISEASE PREDICTION BY OUTLIER ELIMINATION.....</b>	<b>70</b>
Marcin BADUROWICZ <b>DETECTION OF SOURCE CODE IN INTERNET TEXTS USING AUTOMATICALLY GENERATED MACHINE LEARNING MODELS.....</b>	<b>89</b>
Muayed S AL-HUSEINY, Ahmed S SAJIT <b>BREAST CANCER CAD SYSTEM BY USING TRANSFER LEARNING AND ENHANCED ROI.....</b>	<b>99</b>





*Keywords: FEM, composite spar, main rotor blade, carbon fiber, strength analysis*

*Rafał KLIZA* [0000-0002-7571-158]\*, *Karol ŚCISŁOWSKI* [0000-0003-0337-5863]\*,  
*Ksenia SIADKOWSKA* [0000-0002-3315-2102]\*, *Jacek PADYJASEK*\*,  
*Mirosław WENDEKER* [0000-0003-4121-8818]\*

## STRENGTH ANALYSIS OF A PROTOTYPE COMPOSITE HELICOPTER ROTOR BLADE SPAR

### Abstract

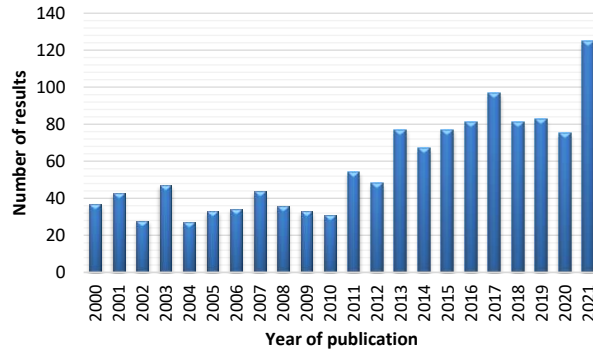
*This paper investigates the strength of a conceptual main rotor blade dedicated to an unmanned helicopter. The blade is made of smart materials in order to optimize the efficiency of the aircraft by increasing its aerodynamic performance. This purpose was achieved by performing a series of strength calculations for the blade of a prototype main rotor used in an unmanned helicopter. The calculations were done with the Finite Element Method (FEM) and software like CAE (Computer-Aided Engineering) which uses advanced techniques of computer modeling of load in composite structures. Our analysis included CAD (Computer-Aided Design) modeling the rotor blade, importing the solid model into the CAE software, defining the simulation boundary conditions and performing strength calculations of the blade spar for selected materials used in aviation, i.e. fiberglass and carbon fiber laminate. This paper presents the results and analysis of the numerical calculations.*

### 1. INTRODUCTION

Composite materials have been recently more and more often applied to aircraft constructions. Constructions of both newly manufactured airplanes and helicopters show a growing share of composite materials. It is worth noting that composite elements are used not only in fuselages but also blades of main rotors. This entails complex loads carried by such a composite material in a main blade. Advanced analytical calculations and numerical simulations become indispensable to correctly design such a blade. Increased interest, especially in 2021, in simulations to design composite blades for helicopter rotor blades was also noticed, which is proven by the publications in the ScienceDirect.com database registered over the last 21 years. The number of publications selected by the keywords of "helicopter composite rotor blade" are listed in Fig. 1.

---

\* Lublin University of Technology, Faculty of Mechanical Engineering, Department of Thermodynamics, Fluid Mechanics and Aviation Propulsion Systems, Lublin, Poland, r.kliza@pollub.pl, k.scislowski@pollub.pl, k.siadkowska@pollub.pl, m.wendeker@pollub.pl



**Fig. 1. Number of publications on helicopter composite rotor blade**

Airfoils are usually made of thin metal skin plating that is constructed to withstand tensile and compressive loads. The nature of such a structure makes it vulnerable to the process of buckling even under low compressive load. Aerospace structures are reinforced by stiffening a load-bearing airfoil with spars which are a main load-bearing element of a rotor blade. A spar is usually a beam which carries loads that are both concentrated and distributed along the blade. Ribs are mounted to a spar later and such a rotor blade supporting structure prevents deforming the support airfoil. Ribs of the airfoil supporting structure transfer loads from the skin to the spar itself (Megson, 2010). The basic feature of a single-spar wing is that only one main spar is along the entire wing, and ribs and baffles make it aerodynamic. A single-spar construction is rarely used in main rotor blades. A very popular method for constructing main rotor blade support structures is honeycomb. The attempts to improve honeycomb structures of rotor blades by increasing strength and stiffness of the entire structure are discussed in several papers (Peng & Bargmann, 2021; Teter & Gawryluk, 2016; Jaafar, Makich & Nouari, 2021; Sukmaji et al., 2017). One way to improve the design of the blade support structure is to make the shape of the structure itself more optimized, which means that, e.g. structures of main rotors are less loaded but their aerodynamic and strength properties remain similar. Defects and methods of their detection in honeycomb structures in plastic rotor blades are classified in several papers (Balaskó et al., 2005; Shahani & Mohammadi, 2015; Rathod, Tiwari & Chougale, 2019; Klochkov et al., 2021). Attention has also been paid to identify typical defects during production of honeycomb blades.

With the increasing popularity of plastics in general industry, the aerospace industry has also become interested in manufacturing such aircraft components. Particular attention has been given to use composites for aerospace applications which account for approximately 70% of the cost of aircraft components. An increasing number of airframe components are being made from non-metallic materials, and one of them is the main rotor spar made more and more frequently of polymer composites. Its greatest advantage is that the weight of such a component is approximately half that of metal alloys, its fatigue strength remains the same and such components can be connected without screws and rivets. Reducing the weight of polymer composite elements is possible due to the low density of materials used in the production of the composite. High strength is achieved by using fiber matrices with high stiffness and strength in the direction of fiber alignment. No screw and rivet connections are possible due to the adhesive properties of composites. The bonding process can occur between two metals, metal and composite, and between two composites (Waghmare et al.,

2021; Kang et al., 2021; Li et al., 2021). An example is bonding the rotor blade skin to ribs and spars inside the blade. Polymer composites used in aviation take one of two forms. The first form is a composite in the form of a laminate where each layer has a different angle of arrangement of fibers, which results in uniform strength in each direction. The second form is a composite of the so-called "sandwich" structure. This structure has a core in its center to provide stiffness to the structure and its outer layers are made of laminate to increase the ability to transfer longitudinal loads (Grodzki, Łukasiewicz & Leśniewska, 2015; Puchała, Jachimowicz & Szymczyk, 2014). The core is bonded to the outer laminate through the so-called adhesive layer. In sandwich composites, the core plays a key role and must provide sufficient stiffness. The effect of core-dependent stabilization of the entire composite is presented in the research paper (Michalski & Krauze, 2019) that focuses on honeycomb cores.

**Tab. 1. Properties of the most common polymer composites**

<b>Parameter</b>	<b>Carbon fibers (50% of volume)</b>	<b>Glass fibers (50% of volume)</b>	<b>Kevlar fibers (type 49)</b>	<b>Graphite fibers (T300/5208)</b>
Density [kg/m <sup>3</sup> ]	1800	2.6	1.4	1.8
Young Modulus [GPa]	230	76	112.4	132
Tensile strenght [GPa]	4.2	3.5	3.0	5.7
Extension (%)	1.9	4.8	2.4	2.8

At present, research is being conducted all the time to improve safety, manufacturing, efficiency, and optimization of polymeric structures. The research paper (Visweswaraiah et al., 2013) focuses on optimization depending on the layer angle and internal geometry of composite materials in a helicopter blade. The publication (Taymaz, 2017) provides information on optimization of composite joints by using innovative algorithms in CAE type programs. Research is also conducted to improve the reliability of composite materials as in the article (Rasuo, 2011) where modern experimental techniques are studied to evaluate the fatigue characteristics of laminated structures. In addition, the publication (Szymański, 2020) presents certain ways of the non-destructive testing of thermoplastic composites made of carbon fibers. The main aspect of the influence of hole preparation in composites on their properties is, in turn, presented in (Karny, 2017).

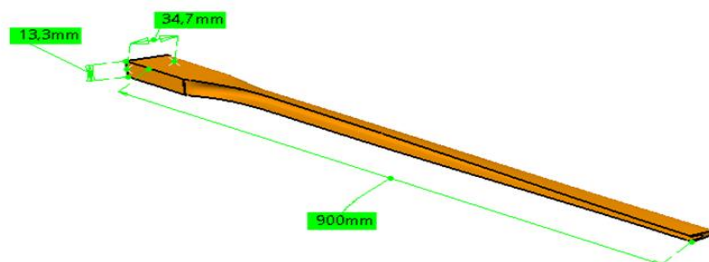
This paper discusses the strength calculations of a prototype main rotor blade spar used in an unmanned helicopter. The calculations were performed with the Finite Element Method and Computer-Aided Engineering software. The strength calculations were performed for the blade spar made of selected materials that are used in aviation.

## **2. MATERIALS AND METHOD**

### **2.1. Preparing of the rotor blade spar model**

The spar was simulated as a static case in ABAQUS. The first calculation step was reduced by the given options to make a more accurate analysis, increase the maximum calculation steps and the minimum increment of the calculation progress. The force application points and the model restraints were determined after the material data were entered to create the 3D model. The support points of the 3D model were determined by

specifying points, lines or planes where the support were be located. Next, the degrees of freedom to be received at the determined locations were selected. The forces acting on the test object were specified just like the support locations were selected, but the type of force acting at a given location was also selected. The type of correlations enables us to modify, e.g. characteristics of friction coefficient between objects or stiffening many parts for a single element. Finally, prior to calculations a model is discretized, which means dividing an object into smaller sections called finite elements. This process requires us to select a correct finite element mesh to show a geometric shape, number of nodes, function of this shape or size of each element. A particularly important parameter is the size of a single finite element. Simulation time becomes longer due to improving accuracy and reducing the size of such an element by maintaining the shape of the test object (Wysmulski, Debski & Falkowicz, 2020; Debski, Rozylo & Wysmulski, 2020; Azad, Mirghaderi & Epackachi, 2021, Różyło & Wrzesińska, 2016). Main rotor simulations in CAE computer programs are typically strength simulations. Numerical calculations are mainly to analyze the deflection of rotor spars and adjacent parts as well as values of maximum stresses at the connection points with other elements. Practical application of the FEM simulation method to obtain preliminary data about the strength of spars with actuators for changing the angle of the entire blade is discussed in the paper (Siadkowska & Borowiec, 2021). In another paper, the simulation was carried out to verify the strength of the model under a given load resulting from flight conditions (Skiba, 2019). The possibility of testing different solutions in the design of holes in the main shaft of the main rotor and the effect of the chosen solution on the maximum stress under a given load was also presented (Skiba et al., 2021). The tested object was a three-dimensional model of the spar, shown below, which was designed using CATIA V5 R21.



**Fig. 2. Spar model with its overall dimensions generated in CATIA V5 R21**

The model shows a spar that can be used in a prototype unmanned helicopter with a maximum mass of up to 150 kg. Ultimately, the smart materials technology is also planned to improve aerodynamic efficiency during each phase of flight. Better aerodynamics was possible by changing the geometry of the main rotor blades so it was decided to perform strength simulations of the spar that is the supporting element of this blade. For this project, a configuration of eight layers of composite was selected and two simulations with a fiber distribution of 0/45/-45/90S and 0/90 were carried out. For clarity, in Tab. 2, abbreviated symbols have been assigned to the individual composite configurations to ease further interpretation of the numerical simulation results.

**Tab. 2. Summary of the composite configurations used for numerical analyses**

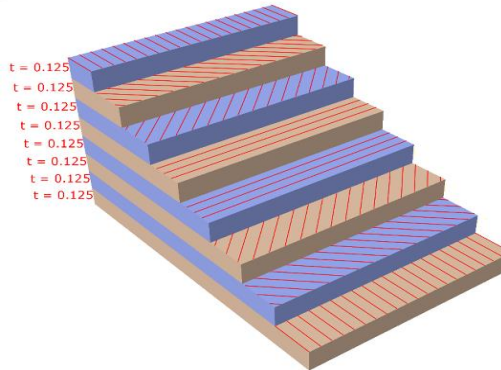
Composite fibre type	Arrangement configuration	Short name
Carbon fiber	0/45/-45/90S	Model 1
Carbon fiber	0/90S	Model 2
Glass fiber	0/45/-45/90S	Model 3
Glass fiber	0/90S	Model 4

The fiber distribution was chiefly parallel to the longitudinal axis of the spar. Unlike isotropic materials, it was required to specify in composites material data for different load directions. The material data determined for the simulations are shown in Tab. 3 both for the carbon fiber composite and the glass fiber composite.

**Tab. 3. Material data with units for carbonfiber/epoxy and fiberglass/epoxy resin composites**

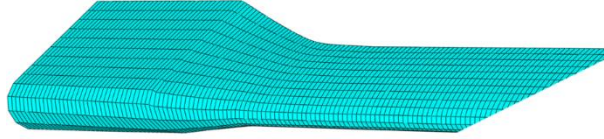
Parameter	Carbonfiber	Fiberglass
Density $\delta$	1800 kg/m <sup>3</sup>	2600 kg/m <sup>3</sup>
Young modulus in paralel direction to fiber arrangement $E_1$	$2.3 \cdot 10^5$ MPa	$7.6 \cdot 10^4$ MPa
Young modulus in perpendicular direction to fiber arrangement $E_2$	$7.5 \cdot 10^3$ MPa	$1.59 \cdot 10^4$ MPa
Poisson ratio in the v fiber plane	0.31	0.25
Kirchhoff modulus G12 in the XY plane	4 510 MPa	5 810 MPa
Kirchhoff modulus G13 in the XZ plane	4 510 MPa	5 810 MPa
Kirchhoff modulus G23 in the YZ plane	3 050 MPa	5 690 MPa
Material type	Laminate	Laminate

The fiber arrangement is illustrated in Fig. 3 for model 1 and 3. Each of the eight layers occupies 12.5% of the laminate volume. All layers have the same thickness.



**Fig. 3. Fiber alignment for model 1 and 3**

A three dimensional tetrahedral mesh was superimposed on the model. This computational mesh consisted of 12 654 elements and 18 370 nodes. The number of elements and the exact type of the finite element mesh is large enough to claim that the simulation calculations are valid and a computation time is short.



**Fig. 4. A spar as a discrete element**

## **2.2. CAE numerical program**

The spar was simulated as a static case using the "Static, General" option in ABAQUS. The loads were defined by introducing boundary conditions in the form of fixing the spar at its cap. Movement along X, Y and Z axes was blocked but rotation was possible around the X axis. Next, the forces acting on spar were defined. The first one was the gravitational force as the "gravity" type in units of  $m^2/s$  being opposite to the Y axis. The force referred to the whole model and its value was distributed uniformly. The centrifugal force to simulate the rotational motion of the spar during flight acts on the axis of the points of [0;0;0] and [0;1;0], i.e. around the Y axis. The centrifugal force was modeled by the function of "Rotational body force" with a centrifugal force effect, and speed was set up as 1200 rpm, or 125.66 rad/s. This force acted uniformly on the entire model. Another component was the lift force generated during the flight of the machine. The induced losses at the blade tip were neglected in the analysis. The lift force acted along the Y axis increasing from the blade cap to its tip. Its value was 500 N, starting from 125 N where the load distribution began. The load distribution was along 750 mm. The load function creation option in ABAQUS was used to simulate this force. Similarly, a drag force estimated as 30 N was introduced starting from 7.6 N where the force originated. The final step was to introduce the spar torsional moment by setting a concentrated moment at the end of the spar of 15 Nm. The torsional moment was distributed uniformly to the plane at the end. A centrifugal force was applied to the blade during the simulated operating conditions, the value of which is calculated from equation (1):

$$F_C = m * \omega^2 * R \quad (1)$$

where:  $m$  – mass,  
 $\omega$  – angular velocity,  
 $R$  – blade length.

The tensile stresses are therefore calculated from the formula:

$$\sigma_b = \frac{F_c}{A} \quad (2)$$

where:  $F_c$  – tensile force,  
 $A$  – cross section of the spar.

In addition to the tension due to the centrifugal force, the spar also carries bending stresses due to the lift force and the drag force. To determine this, start by determining the bending moment from the lifting force in the following formula (3):

$$\sigma_x = \frac{M_{bx}}{W_x} \quad (3)$$

where:  $M_{bx}$  – bending moment against the X axis,  
 $W_x$  – bending strength index also against the X axis.

The bending moment is calculated from the resultant force causing bending and the arm on which it acts relative to where the blade is attached. The resultant force is calculated from the given linear load distribution (4):

$$F_{Lt} = \frac{(F_{Lmax} - F_{Lmin}) * L}{2} + F_{Lmin} * L \quad (4)$$

where:  $F_{Lt}$  – resultant of the lift vector,  
 $F_{Lmin}$  – minimum lift assumed to be 125 N,  
 $F_{Lmax}$  – maximum lift assumed to be 500 N,  
 $L$  – distance over which the lift acts is 750 mm.

The section over which the force acts is determined by the following formula:

$$L_t = \frac{\left(F_{Lmin} * \frac{L}{2}\right) + \left[(F_{Lmax} - F_{Lmin}) * \frac{2}{3} * L\right]}{F_{Lt}} \quad (5)$$

which after substituting the formula for the value of  $F_{Lt}$  gives formula (6).

$$L_t = \frac{\left(F_{Lmin} * \frac{L}{2}\right) + \left[(F_{Lmax} - F_{Lmin}) * \frac{2}{3} * L\right]}{\frac{(F_{Lmax} - F_{Lmin}) * L}{2} + F_{Lmin} * L} \quad (6)$$

Therefore, the bending moment formula can be written as in (7).

$$M_{bx} = F_{Lt} * L_t = \left[\frac{(F_{Lmax} - F_{Lmin}) * L}{2}\right] + (F_{Lmin} * L) * \left\{\frac{\left(F_{Lmin} * \frac{L}{2}\right) + \left[(F_{Lmax} - F_{Lmin}) * \frac{2}{3} * L\right]}{\frac{(F_{Lmax} - F_{Lmin}) * L}{2} + F_{Lmin} * L}\right\} \quad (7)$$

To determine bending stresses, it is necessary to calculate the bending strength ratio with respect to the X axis. Equation (8) defines this ratio as:

$$W_x = \frac{J_x}{e_{max}} \quad (8)$$

where:  $J_x$  – geometric moment of inertia against the bending neutral axis,  
 $e_{max}$  – distance from the bending neutral axis to the outermost point of the cross-sectional contour of the spar.

Thus, the bending stresses from the lifting force will be expressed by equation (9).

$$\sigma_{bx} = \frac{\left[ \frac{(F_{Lmax} - F_{Lmin}) * L}{2} + F_{Lmin} * L \right] * \left\{ \frac{(F_{Lmin} * \frac{L}{2}) + [(F_{Lmax} - F_{Lmin}) * \frac{2}{3} * L]}{\frac{(F_{Lmax} - F_{Lmin}) * L}{2} + F_{Lmin} * L} \right\}}{\frac{J_x}{e_{max}}} \quad (9)$$

Exactly the same formulas apply to blade bending by frictional forces. The difference is the plane of action of the forces so the reference point for the bending formulae is the Y axis.

Hence, to calculate the total bending stress with respect to the X and Y axes, it is necessary to apply formula (10):

$$\sigma_b = \sqrt{\sigma_{bx}^2 + \sigma_{by}^2} \quad (10)$$

where:  $\sigma_b$  – total bending stress,  
 $\sigma_{bx}$  – bending stress against the X axis,  
 $\sigma_{by}$  – bending stress against the Y axis.

Since the blade is also twisted, the total stresses from the force moments are calculated using the Huber hypothesis formula (11):

$$\sigma_{overall} = \sqrt{\sigma_b^2 + 3\tau_t^2} \quad (11)$$

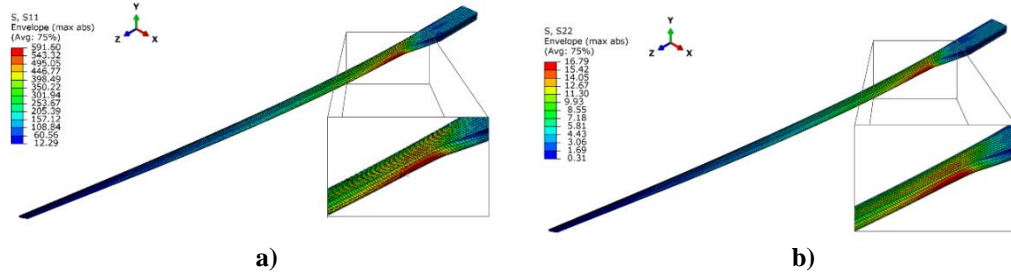
where:  $\sigma_{overall}$  – total stress value,  
 $\sigma_b$  – normal stress from bending moments,  
 $\tau_t$  – tangential stress from torsional moments.

### 3. SIMULATION RESULTS AND DISCUSION

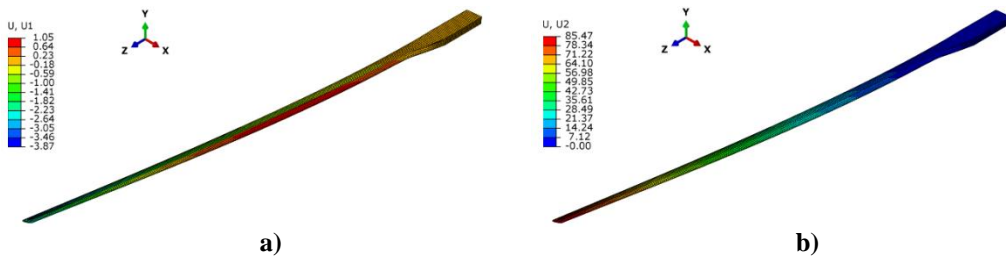
#### 3.1. Composite spar simulation results for model 1

During strength analysis of composites, a key aspect is to determine fiber alignment and stresses acting in each layer. For this reason, loads in two stress states: stresses along the main axis of fiber arrangement (Fig. 5a) and stresses transverse to the main axis of fiber alignment (Fig. 5b) were analyzed in the composite models under consideration. The maximum stress reaches 591 MPa. In the case of stress distribution perpendicular to the main axis of fiber alignment (Fig. 5b), the stresses transmitted through the material are much smaller and reach the highest value of 17 MPa. The strength of carbon fibers perpendicular to the main axis of fiber alignment is estimated to be about 900 MPa, so the material should retain its strength properties at maximum stresses. The deformations along the X-axis (Fig. 6a) reach a maximum displacement of 4 mm opposite to the X-axis. The displacements along in the Y-axis (Fig. 6b) reach a maximum value of 85 mm at the end of the spar.



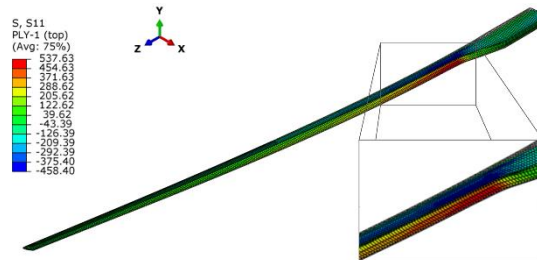


**Fig. 5. Overall stresses for carbon fiber/epoxy resin composite: a) along the main axis of fiber alignment, b) perpendicular to the main axis of fiber alignment.**

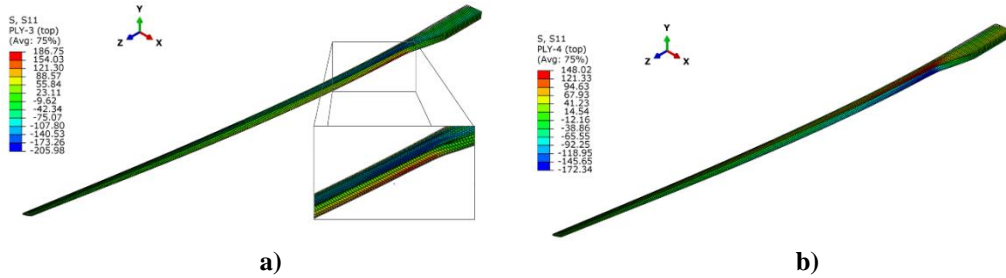


**Fig. 6. Deformations in carbon fiber composite spar: a) along the X-axis, b) along the Y-axis**

The maximum stresses that occurred are longitudinal to the fiber alignment. Due to the laminate structure, the individual stresses in the fibers that occurred in each alignment variant of model 1 were analyzed later in this article (Fig. 7, Fig. 8). The fibers aligned along the main alignment axis show stresses in the range of  $-458 \text{ MPa} \div 538 \text{ MPa}$ . The laminate layer with the fibres aligned at  $45^\circ$  shows lower stresses with the maximum value of  $262 \text{ MPa}$  for fibre tension and  $152 \text{ MPa}$  for fibre compression (Fig. 7b). The stresses in the  $-45^\circ$  laminate layer (Fig. 8a) show differences in stresses with respect to the  $+45^\circ$  laminate, i.e. the highest stresses reached  $187 \text{ MPa}$  for tension and  $205 \text{ MPa}$  for compression.



**Fig. 7. Stresses in the fiber layer along the main axis of fiber alignment**

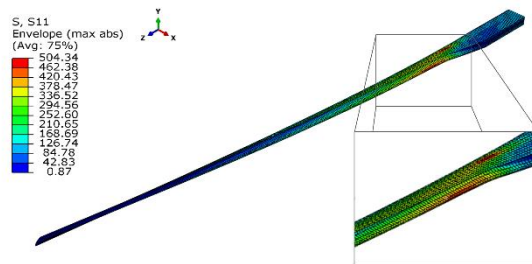


**Fig. 8. Stresses in the layer with fibers: a) angled (-45°) to the main axis of fiber alignment, b) perpendicular (90°) to the main axis of fiber alignment**

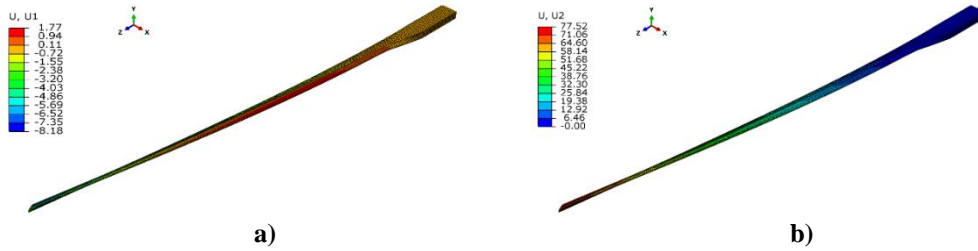
The last layer analyzed was the layer with fibers aligned at the angle of 90° (Fig. 8b). The maximum stresses were 148 MPa for tension and 172 MPa for compression. A significant change compared to the case in Fig. 6 is that compression occurs at the cap of the spar and tension at the tip. The analysis of the results from each separate laminate layer shows that the highest stresses occurred in the layer with fibres parallel to the longitudinal axis of the spar because of the alignment of the fibers taking up the largest portion of the bending stresses that occurred when the load-bearing force was generated in the spar. For the fibers transverse to the first layer, the stresses were much lower and compression and tension occurred on the opposite sides of the spar. This may be due to the greater transfer of load caused by the drag force and the bending moment in the fibers transverse to the longitudinal axis of the spar and a negligible transfer of the bending moment caused by the lifting force.

### 3.2. Composite spar simulation results for model 2

The next variant was a spar made of carbon fiber with epoxy resin, with layers aligned alternately at 0° and 90°. The maximum stresses along the main axis of fiber alignment reached 504 MPa (Fig. 9). The highest stresses occurred on both the bottom and top surfaces. The maximum stresses were by 87 MPa (14%) lower than those in the composite with a uniform fiber alignment. As in the model 1, transverse stresses were significantly lower than longitudinal ones with a maximum value of 15 MPa near the spar cap. This is a difference of 2 MPa compared to the composite with parallel fibers. The model displacements of along in the X axis (Fig. 10a) were more than those of the 0/45/-45/90 composite and the maximum deformation was 8 mm at the spar tip. On the other hand, the displacements along the Y axis (Fig. 10b) were smaller, i.e. 77 mm compared to 85 mm in the composite with fibers arranged uniformly at each angle.

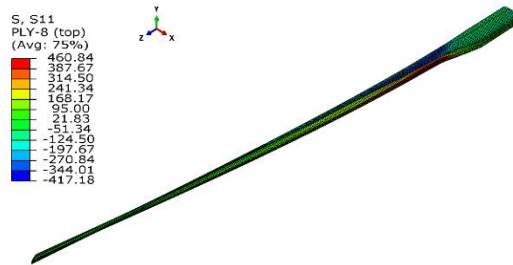


**Fig. 9. General stresses along the main axis of fiber alignment**



**Fig. 10. Deformations in the carbon fiber composite spar: a) along the X axis, b) along the Y axis**

Each layer with fibers aligned at  $0^\circ$  had a similar stress distribution. The maximum values changed, depending on the position of the layer in the composite. Fig. 11 shows the stress distribution for each layer. Tab. 4 shows the individual values of stresses. The first layer is closest to the inner part of the spar, while the eight layer is closer to the outer surface. The stresses in the layers with fibers at the angle of  $90^\circ$  are presented in a similar manner. Tab. 5 shows the stress values in each layer. The simulation shows that the fibers arranged along the spar carry 5 times higher loads than the fibers arranged transversely to the spar because of better properties of transferring the bending load of the spar by the generated load force along its length.



**Fig. 11. Stresses in the fiber layer along the main axis of fiber alignment.**

**Tab. 4. Stress values for the layers with fibers aligned at  $0^\circ$**

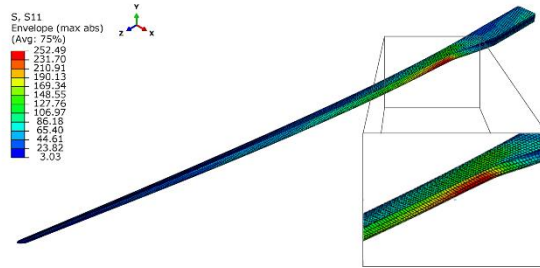
Layer	Tensile stresses [MPa]	Compressive stresses [MPa]
First	400	383
Third	416	392
Sixth	439	404
Eight	455	414

**Tab. 5. Stress values for the layers with fibers aligned at  $90^\circ$**

Layer	Tensile stresses [MPa]	Compressive stresses [MPa]
Second	55	87
Fourth	34	34
Fifth	35	37
Seventh	79	49

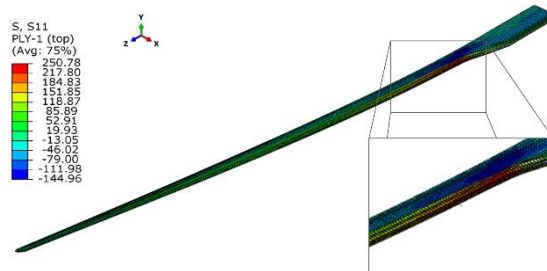
### 3.3. Composite spar simulation results for model 3

A simulation was performed for a composite made of fiberglass and epoxy resin. Fig. 12 shows the overall stresses for the variant according to model 3 and its maximum stress was 252 MPa, which is much lower than in the case of the carbon fiber spar with the same fibers alignment (591 MPa). The stresses transverse to the fiber alignment reached 48 MPa, which is by 31 MPa higher than in the carbon fiber composite with the same fiber alignment.



**Fig. 12. General stresses along the main axis of fiber alignment**

The fibers aligned along the main axis showed stresses of 250 MPa for tension and 144 MPa for compression (Fig. 13). The distribution of stresses was similar to their overall distribution in the spar. For the layer with fibers at  $45^\circ$ , the stresses decreased and were 124 MPa for tension and 42 MPa for compression. The layer with fibers at  $-45^\circ$  showed even lower stresses reaching 86 MPa for tension and 65 MPa for compression of the fibers. The last layer had fibers arranged at  $90^\circ$ , and the stresses here were the lowest of all layers. The tensile stress of the fibers was 47 MPa and the compressive stress was 66 MPa. The fiber compression in this layer occurred at the tip of the spar, and the tension at its cap which is opposite to the other layers.



**Fig. 13. Stresses in the layer with fibers along the main axis of fiber alignment ( $0^\circ$ )**

### 3.4. Composite spar simulation results for model 4

The last variant was a model 4 composite spar. The maximum stresses along the main axis of fiber alignment reached 232 MPa. Fig. 14 shows the stress distribution in the spar. The maximum transverse stresses were 47 MPa, which is 5 times lower than the longitudinal stresses. In the case of separate layers, each layer with a fiber alignment angle of  $0^\circ$  had a similar stress distribution. The maximum values changed, depending on the position of the layer in the composite. Fig. 15a shows the stress distribution for the layer with a  $0^\circ$  fiber alignment angle, while Tab. 6 shows the values in the individual layers. The stresses in each of the layers with a  $90^\circ$  fiber alignment angle are described in a similar manner. Fig. 15b shows the stress distribution in these layers, and Tab. 7 shows their values.

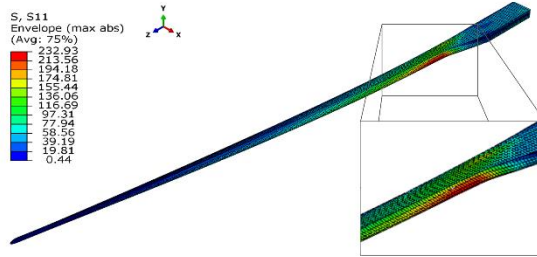


Fig. 14. General stresses along of the main axis of fiber arrangement

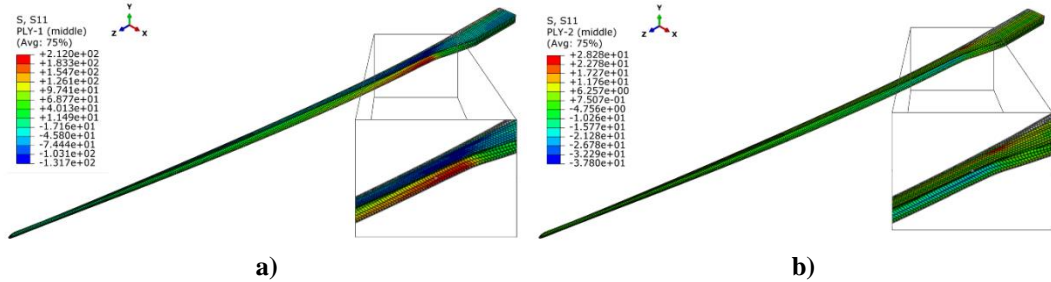


Fig. 15. Stresses in a layer with fibers: a) along the main axis of fiber alignment (0°), b) perpendicular to the main axis of fiber alignment (90°)

Tab. 6. Stress for the layers with a 0° angle of fiber alignment

Layer	Tensile stress [MPa]	Compressive stress [MPa]
First	212	131
Third	217	135
Sixth	226	140
Eight	231	144

Tab. 7. Stress for the layers with a 90° angle of fiber alignment

Layer	Tensile stress [MPa]	Compressive stress [MPa]
Second	28	37
Fourth	25	25
Fifth	24	24
Seventh	27	26

#### 4. CONCLUSIONS

This paper discusses the strength simulation of the spar support rotor of an unmanned helicopter. The research scope included a preparation of a 3D model for ABAQUS software analysis and a strength simulation based on theoretical forces acting on the spar during flight. The selected composite materials were simulated for four fiber alignment variants. The results on displacement and stress do not uniquely determine the behavior of the given model under real loading due to the idealized nature of FEM strength simulations. The stress in the model 1 carbon fiber composite was by 17% higher than in the model 2 alignment (591 MPa vs. 504 MPa). The transverse stress was identical to that in the glass fiber composite, i.e. 16 MPa and was significantly lower than the tension stress. The carbon composite showed significantly different displacement along the X axis than the glass fiber composite. The displacement in the model 1 was 4 mm and as much as 8 mm in the model 2, which is a double deformation difference. The displacements along the Y axis are more similar with

a difference of only 10%. The differences between the fiberglass spar in the configuration of model 3 and 4 are seen in the maximum stresses. The mentioned stresses differ by 8% and reach 252 MPa for the model 3 composite and 232 MPa for the model 4 composite. The fiber alignment had no significant effect on the spar tension values along the Y axis. The tension difference was 4%: the displacements along the X axis were not higher than 14%. The stresses transverse to the spar in each composite fiber alignment variant were significantly smaller than the stresses along the composite. This means that most of the stresses originated from the spar bending induced by the lifting force. The lowest transverse stresses, i.e. 16 MPa were recorded for the carbon fiber composite and the highest, i.e. 48 MPa for the glass fiber composite.

The obtained results enabled the preliminary determination of the rotor blade spar loads for selected fiber arrangement configurations and fiber type. The conclusion drawn from the numerical results is that the values less than the maximum allowable load were achieved for each fiber alignment configuration. The research results show that the most favorable solution is to the model 1. Carbon fibre shows uniform strength in each direction and low loads relative to its ultimate strength. Despite the high excess strength of the spar, its use over many years can lead to fatigue and lower ultimate load values so the model 2 and model 4 variants may be less effective.

## Acknowledgments

*This work has been financed by the National Center for Research and Development under the LIDER IX program. Grant Agreement No. LIDER/45/0177/L-9/17/NCBR/2018.*

## Conflicts of Interest

*The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.*

## REFERENCES

- Azad, S., Mirghaderi, S. R., & Epackachi, S. (2021). Numerical investigation of steel and composite beam-to-encased composite column connection via a through-plate. *Structures*, 31(December 2020), 14–28. <https://doi.org/10.1016/j.istruc.2021.01.040>
- Balaskó, M., Sváb, E., Molnár, G., & Veres, I. (2005). Classification of defects in honeycomb composite structure of helicopter rotor blades. *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 542(1–3), 45–51. <https://doi.org/10.1016/j.nima.2005.01.010>
- Debski, H., Rozylo, P., & Wysmulski, P. (2020). Stability and load-carrying capacity of short open-section composite columns under eccentric compression loading. *Composite Structures*, 252, 112716. <https://doi.org/10.1016/j.compstruct.2020.112716>
- Grodzki, W., Łukasiewicz, A., & Leśniewska, K. (2015). Modelling of UAV'S Composite Structures and Prediction of Safety Factor. *Applied Computer Science*, 11(3), 67–75.
- Jaafar, M., Makich, H., & Nouari, M. (2021). A new criterion to evaluate the machined surface quality of the Nomex® honeycomb materials. *Journal of Manufacturing Processes*, 69, 567–582. <https://doi.org/10.1016/j.jmapro.2021.07.062>

- Kang, Z., Shi, Z., Lei, Y., Xie, Q., & Zhang, J. (2021). Effect of the surface morphology on the bonding performance of metal/composite hybrid structures. *International Journal of Adhesion and Adhesives*, 111, 102944. <https://doi.org/10.1016/j.ijadhadh.2021.102944>
- Karny, M. (2017). The influence of the fastener hole preparation method on the fastener pull-through process in a carbon composite. *Transactions on Aerospace Research*, 1(246), 45–53. <https://doi.org/10.2478/tar-2017-0005>
- Klochkov, N., Zverkov, I., Kurlaev, N., & Ahmed, M. S. (2021). Improvement of non-destructive testing methods in diagnostics of composite honeycomb structures of civil aircraft. *AIP Conference Proceedings*, 2402, 020045. <https://doi.org/10.1063/5.0071712>
- Li, X., Wang, B., Xu, D., Wang, B., Dong, W., & Li, M. (2021). Super-high bonding strength of polyphenylene sulfide-aluminum alloy composite structure achieved by facile molding methods. *Composites Part B: Engineering*, 224, 109204. <https://doi.org/10.1016/j.compositesb.2021.109204>
- Megson, T. H. G. (2010). Introduction to Aircraft Structural Analysis. In *Introduction to Aircraft Structural Analysis*. Elsevier. <https://doi.org/10.1016/C2009-0-62169-3>
- Michalski, M., & Krauze, W. (2019). Influence of honeycomb core stabilization on composite sandwich structure geometry. *Transactions on Aerospace Research*, 3(256), 1–13. <https://doi.org/10.2478/tar-2019-0013>
- Peng, X. L., & Bargmann, S. (2021). A novel hybrid-honeycomb structure: Enhanced stiffness, tunable auxeticity and negative thermal expansion. *International Journal of Mechanical Sciences*, 190, 106021. <https://doi.org/10.1016/j.ijmecsci.2020.106021>
- Puchała, K., Jachimowicz, J., & Szymczyk, E. (2014). Analysis of load transfer into composite structure. *Applied Computer Science*, 10, 86–94.
- Rasuo, B. (2011). Experimental Techniques for Evaluation of Fatigue Characteristics of Laminated Constructions from Composite Materials: Full-Scale Testing of the Helicopter Rotor Blades. *Journal of Testing and Evaluation*, 39(2), 237–242. <https://doi.org/10.1520/JTE102768>
- Rathod, S., Tiwari, G., & Chougale, D. (2019). Ballistic performance of ceramic-metal composite structures. *Materials Today: Proceedings*, 41, 1125–1129. <https://doi.org/10.1016/j.matpr.2020.08.759>
- Różyło, P., & Wrzesińska, K. (2016). Numerical Analysis of Buckling and Critical Forces in a Closed Section Composite Profile. *Applied Computer Science*, 12(2), 54–62.
- Shahani, A. R., & Mohammadi, S. (2015). Damage tolerance approach for analyzing a helicopter main rotor blade. *Engineering Failure Analysis*, 57, 56–71. <https://doi.org/10.1016/j.engfailanal.2015.07.025>
- Siadkowska, K., & Borowiec, P. (2021). Strength analysis of the conceptual model of a main rotor blade spar with actuators. *Journal of Physics: Conference Series*, 1736(1), 012021. <https://doi.org/10.1088/1742-6596/1736/1/012021>
- Skiba, K. (2019). Designing and FEM simulation of the helicopter rotor and hub. *IOP Conference Series: Materials Science and Engineering*, 710, 012003. <https://doi.org/10.1088/1757-899X/710/1/012003>
- Skiba, K., Raczynski, R., Kliza, R., & Wendeker, M. (2021). Strength analysis of a propulsion shaft dedicated for the main rotor test bench. *Journal of Physics: Conference Series*, 1736(1), 012052. <https://doi.org/10.1088/1742-6596/1736/1/012052>
- Sukmaji, I. C., Wijang, W. R., Andri, S., Bambang, K., & Teguh, T. (2017). Application of sandwich honeycomb carbon/glass fiber-honeycomb composite in the floor component of electric car. *AIP Conference Proceedings*, 1788, 030056. <https://doi.org/10.1063/1.4968309>
- Szymański, R. (2020). Non-destructive testing of thermoplastic carbon composite structures. *Transactions on Aerospace Research*, 1(258), 34–52. <https://doi.org/10.2478/tar-2020-0003>
- Taymaz, H. A. (2017). Optimization of Composite Couplings in Helicopter Rotor Blade Spar Using Hybrid Particle Swarm-Gradient Algorithm. *Bilge*, 1(2), 71–78.
- Teter, A., & Gawryluk, J. (2016). Experimental modal analysis of a rotor with active composite blades. *Composite Structures*, 153, 451–467. <https://doi.org/10.1016/j.compstruct.2016.06.013>
- Visweswaraiah, S. B., Ghiasi, H., Pasini, D., & Lessard, L. (2013). Multi-objective optimization of a composite rotor blade cross-section. *Composite Structures*, 96, 75–81. <https://doi.org/10.1016/j.compstruct.2012.09.031>
- Waghmare, S., Shelare, S., Aglawe, K., & Khope, P. (2021). Materials Today: Proceedings A mini review on fibre reinforced polymer composites. *Materials Today: Proceedings*, in press. <https://doi.org/10.1016/j.matpr.2021.10.379>
- Wysmulski, P., Debski, H., & Falkowicz, K. (2020). Stability analysis of laminate profiles under eccentric load. *Composite Structures*, 238, 111944. <https://doi.org/10.1016/j.compstruct.2020.111944>

*Keywords: breast cancer CNN, loss, accuracy, precision, confusion matrix*

Kevin Joy DSOUZA [0000-0002-4103-6718]\*, Zahid Ahmed ANSARI\*\*

# HISTOPATHOLOGY IMAGE CLASSIFICATION USING HYBRID PARALLEL STRUCTURED DEEP-CNN MODELS

## Abstract

*The healthcare industry is one of the many out there that could majorly benefit from advancement in the technology it utilizes. Artificial intelligence (AI) technologies are especially integral and specifically deep learning (DL); a highly useful data-driven technology. It is applied in a variety of different methods but it mainly depends on the structure of the available data. However, with varying applications, this technology produces data in different contexts with particular connotations. Reports which are the images of scans play a great role in identifying the existence of the disease in a patient. Further, the automation in processing these images using technology like CNN-based models makes it highly efficient in reducing human errors otherwise resulting in large data. Hence this study presents a hybrid deep learning architecture to classify the histopathology images to identify the presence of cancer in a patient. Further, the proposed models are parallelized using the TensorFlow-GPU framework to accelerate the training of these deep CNN (Convolution Neural Networks) architectures. This study uses the transfer learning technique during training and early stopping criteria are used to avoid overfitting during the training phase. these models use LSTM parallel layer imposed in the model to experiment with four considered architectures such as MobileNet, VGG16, and ResNet with 101 and 152 layers. The experimental results produced by these hybrid models show that the capability of Hybrid ResNet101 and Hybrid ResNet152 architectures are highly suitable with an accuracy of 90% and 92%. Finally, this study concludes that the proposed Hybrid ResNet-152 architecture is highly efficient in classifying the histopathology images. The proposed study has conducted a well-focused and detailed experimental study which will further help researchers to understand the deep CNN architectures to be applied in application development.*

---

\* Research Scholar (Dept. of CSE, PACE Mangalore), India, joydsouza33@gmail.com

\*\* Professor (Dept. of CSE, PACE Mangalore), India, zansari4@gmail.com



## 1. INTRODUCTION

Medical technology professionals must work hard at developing durable and reliable classification techniques (Aziz, 2017). In the line of healthcare, data and image analysis must be a swift procedure to make sure that speedy diagnoses can ease the lives of numerous patients (Liang et al., 2021). Moreover, this would save the time of nurses and doctors when it comes to diagnosing minor diseases so as they can pay attention to more serious ones and investigate them. Progressive solutions for data and image analysis would also provide healthcare centers with the ability to generate treatment plans quickly. With instantaneous access to different databases and information sources, medical professionals are capable of raising success rates significantly as well as broadening the scope of available treatment options. By the means of advanced classification techniques for the analysis of healthcare data (Pramanik, Pal, Mukhopadhyay & Singh, 2021), professionals are capable of gaining insight into how they can generate precise and budget-friendly healthcare that is also of high caliber. It is with the assistance of such advanced artificial intelligence (Dwivedi et al., 2021) that early diagnosis can take place without leaving the patient unattended to and neglected until their symptoms worsen. High-tech classification techniques will also delve into meta-heuristic algorithms, learning schemes, mobile telemedicine clustering techniques, and brain-computer interfaces. These elements all group together to provide the highest quality patient care programs that are both cost and time-efficient for healthcare staff and customers. By creating a more responsive system of diagnosis and treatment solutions, the healthcare industry would reap the benefits of quick and easy healthcare delivered to all. Artificial intelligence (AI) (Dwivedi et al., 2021) technologies are especially integral and specifically deep learning (DL) (Djellali, Adda & Moutacalli, 2020) a highly useful data-driven technology. It is applied in a variety of different methods but it mainly depends on the structure of the available data. However, with varying applications, this technology produces data in different contexts with particular connotations. For instance, taking spatially correlated data as an example, it includes multiple imaging modalities that could range from radiology to pathology. Not just that but spatially correlated data exists in other forms such as patient monitoring systems that can be defined as electrocardiography (ECG) (Moon et al., 2004) or electroencephalography (EEG).

The future of cancer treatments would be completely revolutionized by the automated diagnosis that utilizes deep neural networks as classifier models as well as recursive feature elimination as a form of feature selection. The spatial-temporal features play a major role in the visual recognition task. CNN's are a very powerful source for the representation of classification results. These deep neural network models can directly extract the features from raw data. The convolution layers in CNN are designed with varying numbers of kernels and a well-suited stride to learn the features from the image. Further, the normalization layers improve data generalization and reduce overfitting. Activation function plays a significant role in classification task and pooling layers reduce the local convolution features complexity. Deep neural networks have a higher footing over other similar classification models such as SVM (Kaur, Singh & Kaur, 2019) since they work with multiple layers. Consequently, the multiple layers are administered to codify the breast cancer data and provide quick, automated, and correct results about it. Deep neural networks have previously been investigated in the Wisconsin Breast Cancer Dataset (UCI Machine Learning Repository, n.d.) from the UCI Machine Learning Repository

where the data was divided into distinct batches of a train-test split. Consequently, the system's functioning was measured upon the basis of precision, recall, specificity, sensitivity, and accuracy. However, deep neural networks are highly computation intensive, which require a lot of time in training depending on the complexity of the network and training data size (Eelbode, Sinonquel, Maes & Bisschops, 2021). By keeping these critical points in mind, the proposed research work uses different deep CNN architectures such as RESNET (Sarwinda, Paradisa, Bustamam & Anggia, 2021) with 50,101,152 layers, VGG16 (Guan et al., 2019), and Mobile Net to propose a hybrid model. While using this model we obtain feature output from the CNN model by removing the topmost classifying layer which provides the vector of features. The spatial features obtained are passed to the LSTM network for extracting the temporal features. LSTM as an advancement to the existing architecture makes it more suitable for the data with nature under consideration. With this set up this study aims to propose a transfer learning-based hybrid classification model which can work efficiently and effectively in the classification of cancer images. To accelerate the training time and computing power the proposed models utilize GPU computing using parallel computing library of tensorflow (TensorFlow Framework & GPU Acceleration. NVIDIA Data Center, n.d.) framework. The hybrid models presented here are evaluated for best performance and it is observed that ResNet152 surpasses the other model with the best performance. The model built here is developed in such a way that it can be trained for any type of microscopic cancer data set. Overall, the remainder of the paper is organized as follows: section 2 gives the brief methodology literature in constructing deep neural architectures. The proposed methodology and experimental results are presented in sections 3, 4, and 5 respectively. Finally, the last section concludes the study.

## 2. LITERATURE SURVEY

In the recent era, breast cancer diagnosis from pathological images has got great attention from the research community. Kiranpreet Kaur et al (Kaur & Mittal, 2020) proposed the technique for extracting spatial-temporal features from the image for cancer detection. It is very important to understand which architecture may be suitable for data in consideration, before jumping to the conclusion. Hence this research work tried to understand the best deep CNN (Convolution Neural Networks) architectures for histopathological image classification. The mentioned work started with considering the pre-trained architectures such as ResNet, MobileNet, VGG16 and trained these models on the considered dataset. The design architecture generates multiple channels of data from input images and convolution is performed. The final feature representation is from all input data. The convolution architecture applied a group of hardware kernels to come up with feature information and encodes previous knowledge on features that results in high performance. The study about pre-trained CNN models proposed by Vijaya Gajanan et al. (Buddhavarapu & Jothi, 2020) proposes transfer learning techniques for architectures such as VGG Net, Inception Net, and ResNet to classify histopathology images. Convolutional neural networks are a class of feed-forward neural networks predominantly used for image processing. The CNN is a class of feed-forward networks with the input layer, convolution layer, pooling layer, fully connected layer, and output layer. The transfer learning technique uses the knowledge obtained by a deep learning model while solving the given

problem, which is transferred to solve related target problems. Generally, the weight of the deep learning model is initialized randomly during training the network from scratch. The transfer learning technique is favorable in two forms. First, when transfer learning is used as a feature extractor, this will be used in solving the target problems with little modification to retain the convolution and pooling layers of the pre-trained model and remove the final fully connected layer and the classifier layer of the pre-trained model. The resultant is called the feature extractor model. The feature extracted here is then fed to any traditional classification. It is also possible to add new fully connected layers and classifier layers here. Further, the study from Said Boumaraf et al. (Boumaraf, Liu, Zheng, Ma & Ferkous, 2021) proposed a new transfer learning model for the classification of histopathological images. In this work, the researcher proposed a technique for intrinsic feature learning using ResNet with a total of 18 layers. This study utilized these pre-trained models to implement the ResNet (Sarwinda et al., 2021) with 52,101 and 152 architectures. Further, the arrangement of layers in other architectures such as VGG16 and MobileNet varies with the number of convolution layers and resultant parameters. We further investigated the structure of these models to be able to use them for the methodology performed in this work. Qian Xiang et al. (Xiang et al., 2019) proposed a classification model based on a lighter model using MobileNet architecture. In this work, a transfer learning technique was used for fruit image classification using the Imagenet dataset. This also discussed strategies to replace the top layer of the base network with the convolution layer and softmax classifier. Also, this work also presented scenario about adding drop out to the newly added conv2d to reduce the overfitting.

Further, it is also studied that the early stopping (Leen, Dietterich & Tresp, 2001) technique is highly useful to control the overfitting of the designed model. Research work VGG16 (Simonyan & Zisserman, 2015) claimed to use smaller 3X3 filters in each convolution layer and combining it in a sequence of convolution. In VGG the configuration in case of depth increases significantly from left to right as per the study stated by Karen Simonyan et al. (Simonyan & Zisserman, 2015) as more layers are added. Along with this the convolution layer parameters are denoted as Conv.

The table 3 below shows the overall comparison of the different architectures.

**Tab. 3. Comparison of CNN Networks**

Comparison					
Network	Year	Salient Feature	top5 accuracy	Parameters	FLOP
AlexNet	2012	Deeper	84.70%	62M	1.5B
VGGNet	2014	Fixed-size kernels	92.30%	138M	19.6B
Inception	2014	Wider – Parallel kernels	93.30%	6.4M	2B
ResNet-152	2015	Shortcut connections	95.51%	60.3M	11B

This study is motivated by aforesaid literature to propose a better deep neural network architecture that is hybrid in design. It also observed that objects in an image are often related to each other. Images are however not sequential but carry some dependencies which can helpful in modeling them as a sequence of occurrences of certain objects. So these latent semantic dependencies can be modeled as a sequence of occurrences using

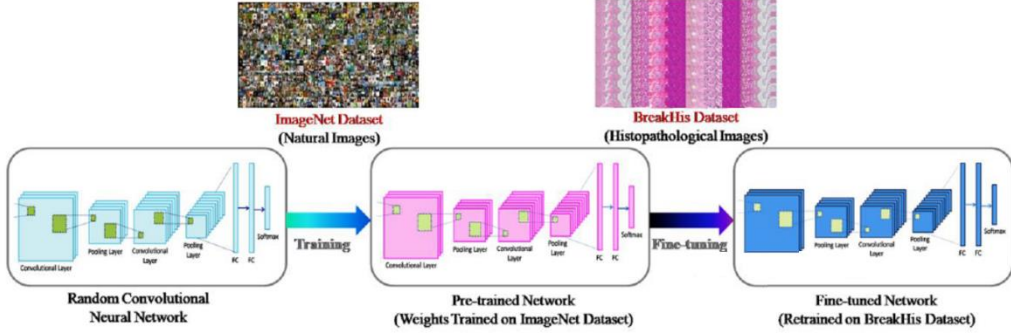
the LSTM network. As added proof, the work presented by SuvidhaTripathi et al. (Tripathi, Singh & Lee, 2021) proposed a bi-LSTM model for the classification of breast tumors. It is also important to note the parameters that a network uses and the time it takes to represent the training data from the computational point of view. The core computing jobs of deep learning are matrix operations, and hence it is possible to turn this into many simple computing jobs. GPU (Haghighat & Juanes, 2020) as they contain many cores have got greater attention in training the deep learning models. General-purpose processors like CPUs have complex instruction systems and execute the instructions in the sequence which is good at processing a small amount of data. High-density computing jobs like deep learning data are complex and as the training data increases the model accuracy can be improved. The work presented by Ehsan Haghighat et al. (Haghighat & Juanes, 2020) was highly motivational to use Tensorflow (TensorFlow Framework & GPU Acceleration. NVIDIA Data Center, n.d.) GPU framework (Deep Learning Frameworks. NVIDIA Developer, n.d.) in our work in implementing these models in our study.

### 3. PROPOSED METHODOLOGY

This study proposes a deep learning model for classifying the histopathological images using intrinsic feature learning techniques based on the Residual CNN, VGG16, and MobileNet architectures. These architectures are pre-trained on ImageNet and this Pre-trained knowledge is used as the backbone model in our approach. The proposed approach firstly normalizes the dataset images, followed by this a three-fold data augmentation is applied on the train set. Following this, we apply the transfer learning method based on the block-wise fine-tuning strategy to transfer the learned from the ImageNet dataset to the medical image data set used in this work. The research work presented here basically modifies the existing architecture by adding a parallel LSTM layer to build a hybrid network. This study Proposes hybrid LSTM model architectures for ResNet, MobileNet, and VGG16. Experimental setup and methodology used are briefly discussed in following sections.

#### 3.1. Transfer Learning

This study uses Transfer Learning (TL) (Zhuang et al., 2021) technique to address the challenge of requiring millions of images to train a deep learning model. In TL technique a network is trained on a base dataset to learn general features of image. The learned features are then transferred to another network to train the network on a target cancer dataset (Figure 2). In this phase the weights of pre-trained networks are transferred to the network to fine-tune it for a complex dataset of histopathological images. The approach is utilized by fine-tuning the layers of the network. All the layers are retrained on the target dataset and the initialization of weights has been carried out from the pre-trained weights in fine-tuning. Using this technique the proposed work handles the scarcity of the dataset and high computational cost (Boumaraf et al., 2021). The layer specialization, size, and complexity of the dataset are some factors that need to be considered while implementing transfer learning (Sertolli, Ren, Schuller & Cummins, 2021) which carefully considered in this phase.



**Fig. 2. Illustration of the transfer learning (TL) approach**

The complexity of the dataset is a major concern as it affects the generalization of acquired information during the training process. The influence of layer-wise fine-tuning of a pre-trained network on the performance of the model on BC histopathological images has been demonstrated using the proposed approach. The experiment is performed by employing VGG16+LSTM, Mobilenet+LSTM, and Resnet152+LSTM using the BreakHis dataset. It is important to emphasize that the aim is not to achieve state-of-the-art performance but to explore the influence of knowledge transfer from natural images (simpler dataset) to histopathological images (complex dataset). Moreover, this work is based on an assumption that the transferring of weights from natural images to histopathological images would lead to better performance, irrespective of their domain.

### 3.2. Training Protocol

The network is trained using a backpropagation algorithm (Improving the convergence of back-propagation learning with second-order methods – NYU Scholars, n.d.) to minimize the cost function ( $C$ ) concerning the weights ( $W$ ), defined as:

$$c = -\frac{1}{|m|} \sum_i^{|m|} \ln(p\left(\frac{n^i}{m^i}\right)) \quad (1)$$

Where  $|m|$  implies the number of images in the training set,  $n^i$  represents the label correspond to the  $i^{\text{th}}$  image of the training set and  $p(n^i/m^i)$  denotes the probability of  $m^i$  to be classified correctly. The cost function is optimized by the stochastic gradient descent (SGD) method which is a powerful tool of optimization in machine learning. SGD considers only a few training samples to follow either the negative or positive gradient of the cost instead of the entire training set and leads to fast convergence at a minimum cost. Consider,  $C$  as the cost over a mini-batch of size  $X$  and  $W_{je}$  as the weights in the  $j$ th convolutional layer at  $e$  epoch (iteration). The updated weights in the next epoch are computed as:

$$\gamma^e = \gamma \left\lfloor \frac{e^N}{|m|} \right\rfloor \quad (2)$$

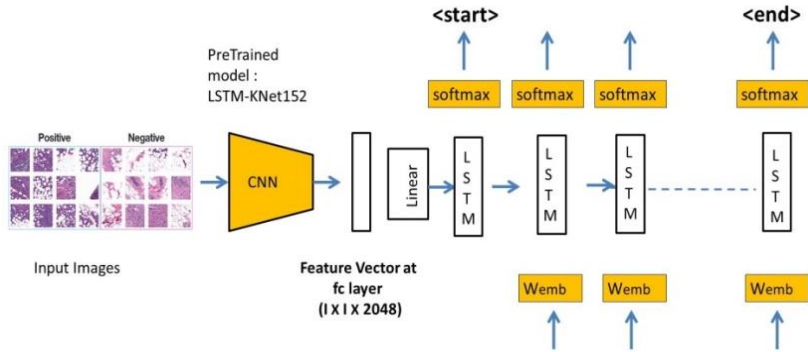
$$V_j^{e+1} = \mu V_j^e - V^t \alpha_j \frac{\partial \mathcal{L}}{\partial w_j} \quad (3)$$

$$W_j^{e+1} = W_j^e + V_j^{e+1}$$

where  $\gamma$  represents the scheduling rate,  $\mu$  denotes the momentum, and  $\alpha_j$  represents the learning rate of the  $j$ th layer. The scheduling rate is used to reduce the learning rate according to a pre-defined schedule with the progression of training, while momentum is a moving average of gradients and is used to update the weights in the network. The value of all these parameters are adjusted during the experiment by an extensive set of trial and found 0.001, 0.9, and 0.95 as a reasonable choice for the learning rate, momentum, and scheduling rate, respectively. A batch of size 500 along with 75,150,200 numbers of epochs per layer is considered for the training process.

### 3.3. Proposed framework

The proposed models are developed using the pre-trained CNN with LSTM layers for feature vector extraction on medical dataset. Objective of combining LSTM is to sustain sequence prediction as presented in Figure 3.



**Fig. 3. Illustration of the overall design of proposed models using parallel LSTM layer**

The pre-trained models specifically VGG 16, MobileNet, ResNet152, Resnet101, and Resnet50 are used for feature vector extraction. They were trained by quite 1.3 million images obtained from the ImageNet (Deng et al., 2010) database. ImageNet is one of the highest datasets in DL and is right to be used within the pre-training part of the model, due to its popularity and data diversity. Here the feature yield from the CNN model is obtained by removing the topmost classifying layer which supplies a vector of features. Here the feature yield from the CNN model is first obtained by removing the topmost layer which provides a vector of features. The spatial features we obtained from the CNN model are passed to the LSTM architecture for extracting temporal features to pass to the trained model for action classification. At Last, a softmax classifier is added to construct the prediction of those high-level features.

The proposed hybrid models are tested for best suited deep learning architecture to classify the histopathology images. The following section will reflect a detailed experimental setup used to evaluate the performance of the model. As it is evident from the results the proposed resnet152 architecture with LSTM network outperformed among other models and hence it is proposed with the name LSTM-KNet152 architecture from this study.

### 3.4. Model parallelism using TensorFlow

Long Short-Term Memory is a powerful network to solve the problem of long-term dependence in the recurrent neural network. This has got attention because of its ability to refine internal processing units to store and update context information efficiently. At each time LSTM takes the input value  $x_t$ ,  $h_{t-1}$  at the previous time and a value  $C_{t-1}$  at the final time. Here the output is the current time  $h_t$  with current state  $C_t$  respectively. Therefore the forget gate  $f_t$  will determine the information that cell discards using the equation:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (4)$$

Here  $\sigma$  represents activation function,  $W$  and  $b$  are weight and bias respectively. The forget gate  $f_t$  outputs the value '0' or '1' by taking  $x_t$  and the cell state  $h_{t-1}$  of the previous moment. Here '1' means TRUE and '0' means FALSE. Then the cell will predict the value to be updated using:

$$i_t = (W_i[h_{t-1}, x_t] + b_i) \quad (5)$$

$$C_t^{\sim} = \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (6)$$

The sigmoid layer determines the value to be updated. A tanh layer will create a new candidate value vector. The status is then updated by cell from  $C_{t-1}$  to  $C_t$  thereby the output  $h_t$ :

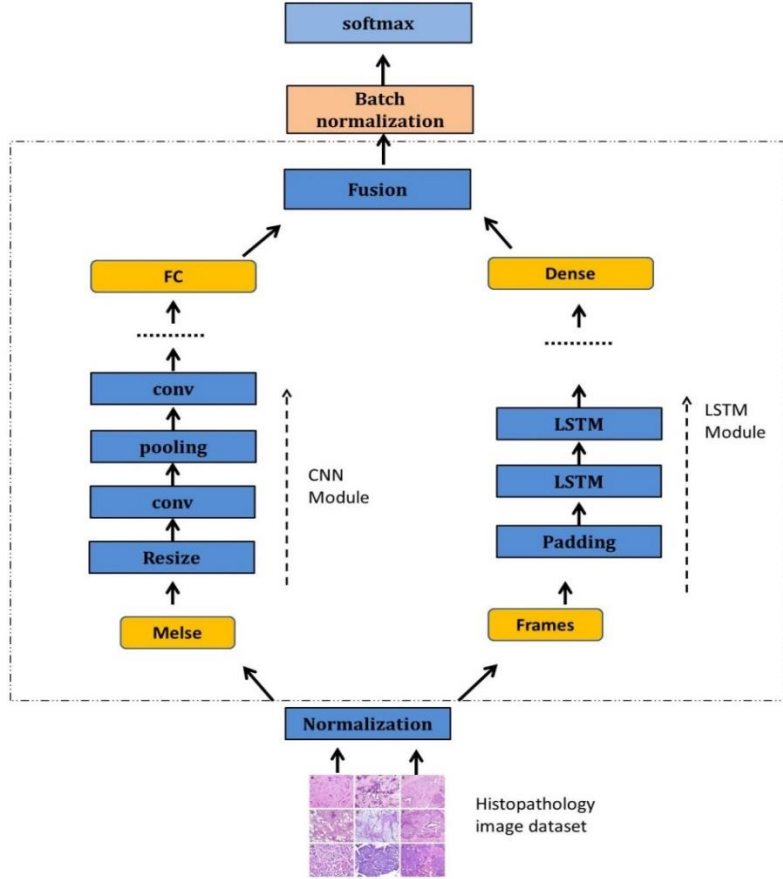
$$C_t = f_t * C_{t-1} + i_t * C_t^{\sim} \quad (7)$$

$$h_t = o_t * \tanh(C_t) \quad (8)$$

The expanded LSTM model can obtain repetitive network structure and share parameters among each network, the parameters of training are relatively reduced and the model can be extended to different length sequences. Therefore this allows the LSTM model to be used for sequences of varying length.

The network structure proposed in this study is described in Figure 4. The main objective is to improve the convergence speed of the model. The network presented here is designed to have a parallel LSTM module to improve the learning thereby promising to produce better performance. In this study, we first normalize the input images. Followed by this we extract two different feature representations using these two parallel modules. First, the CNN module is employed to learn the details of features in histopathology cancer images. To get a better training effect we make use of pre-trained CNN architectures such as MobileNet, VGG-16, ResNet-101, and ResNet-152. Further from which we feature the

Resnet with 150 layers with the power LSTM module outperforms all the remaining architectures. Hence from this study developed a hybrid model we term LSTM-KNet152. The model structure of CNN is shown in Figure 4.



**Fig. 4. Proposed parallel model structure**

There are five convolution layers, three pooling layers, and two fully connected layers in all proposed model structures. Since most convolutional neural networks need fixed inputs size, in this work we will resize images  $Mels \in R^{T \times F \times C}$  into  $227 \times 227 \times 3$ . Here the resize operation is performed using bilinear interpolation. During the same time model uses LSTM to learn the temporal changes of cancer images. LSTM model can handle variable length features by feeding it image features. To accomplish the requirement of the model, these features should be zero-padded into the same dimension. When the time steps are outside the range of actual data length, the LSTM internal parameters stop updating. The model structure of LSTM is shown in Figure 4. Followed by this, we integrate two different types of high-level features. To maximize the convergence speed and avoid gradient diffusion of training, the output was normalized by batch normalization. Finally, a softmax classifier is used to classify histopathology images into benign or malignant.



#### 4. PERFORMANCE EVALUATION

The experiment in this study is conducted by using Breast Cancer Histopathological Database (BreakHis) (Spanhol, Oliveira, Petitjean & Heutte, 2016). This database consists of 9,109 microscopic images of breast tumor tissue collected from 82 patients using different magnifying factors (40X, 100X, 200X, and 400X). As per the present update, this database contains 2,480 benign and 5,429 malignant samples (700X460 pixels, 3-channel RGB, 8-bit depth in each channel, PNG format). The proposed methodology in our experiment is evaluated based on transfer learning techniques applied to popular architecture such as MobileNet with LSTM, VGG-16 with LSTM, ResNet-101 with LSTM, and ResNet-152 with LSTM. This study is conducted to evaluate the best architecture model to be proposed for histopathology cancer image classification. The performance of the classifier for the transfer learning approach is examined using accuracy, precision, recall, F1-score, and support. Further, to make the model learn the input information over sufficient large iterations of the epoch, model performance is evaluated for a set of 150, 200 epochs. Further, to address the overfitting issue we have implemented early stopping criteria where all considered models produced the best performance. For each set of the experiment, the considered dataset is divided into 90%:10% for training and testing respectively, and the training set is further divided into 80% for training purposes and 20% for validation purposes. The splitting of the dataset is done in a manner that the patients involved in the building of the training set are not included in the testing set. In this study, we use a parallel computing framework to address the computational complexity imposed by these deep CNN architectures. We use NVIDIA RTX2060 GPU combined with the power of the TensorFlow-GPU framework to accelerate the training speed. This GPU has 1,920 CUDA cores with 240 tensors that can deliver 52 teraflops of deep learning horsepower. This is a powerful addition to our developed network to make it further trained in parallel using these multiple cores. Our effort in this study at the same time lies in proposing a highly parallel training technique applied to accelerate the training time of this deep architecture on a single GPU. This highly favorable in the cost of resources required to implement these deep architectures.

This study proposes the novel deep learning architecture using well-known existing architecture like VGG-16, Mobile Net, ResNet-101, and ResNet-152. Existing architecture are reused by combining them with LSTM network to form a hybrid model. The contribution of this work is to present the best-suited hybrid architecture for the classification of pathology cancer images. Even though the model is pre-trained using the Imagenet database, in order to make model to learn cancer image features the network is fine-tuned using the BreakHis data set. Further, it also noted that the developed model is efficient to classify any random new microscopic image provided related features are modeled in advance. During initial trial, experiment utilizes 2000 images for validation. The deep learning model is trained for different values of epoch and the performance of the model is recorded. The following tables depict the overall performance of the models for different epoch values; it is evident that the early stopping technique is significant in solving the overfitting problem. Hence the models produce the best performance with the learning rate noted.

## Hybrid MobileNet Model Performance Comparison

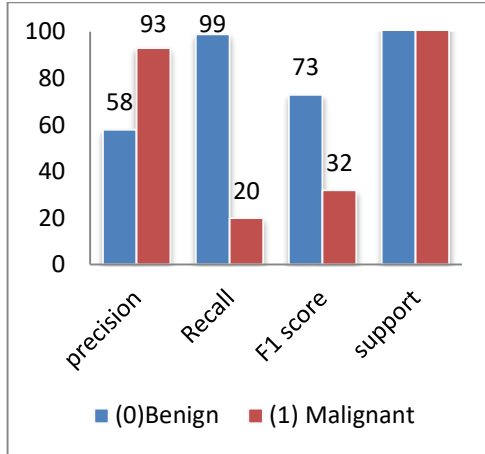


Fig. 5. model Performance for epoch 100

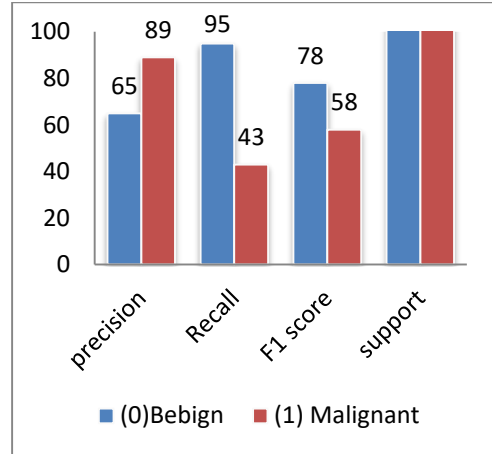


Fig. 6. model performance for epoch 200

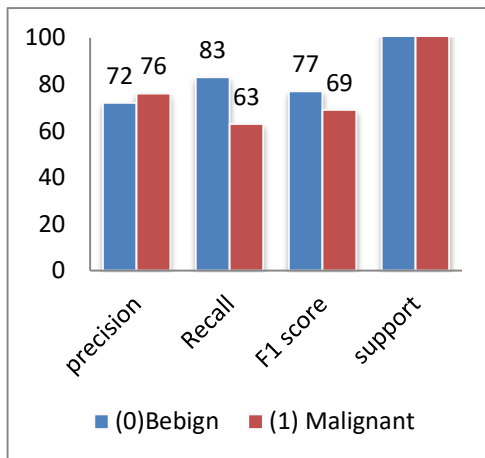


Fig. 7. early stopping at epoch 43

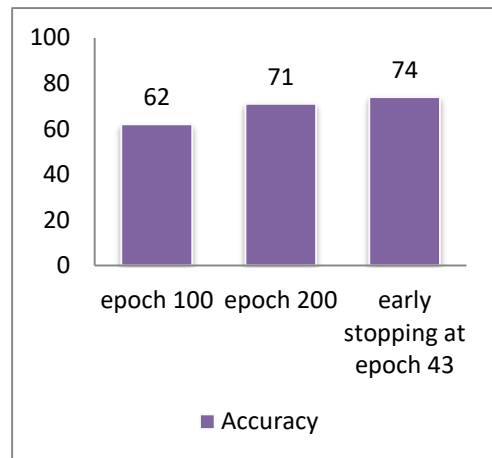


Fig. 8. Accuracy of model

To follow the best practice of training the model, experiments are conducted for epoch trials 100 and 200. To avoid overfitting, early stopping criteria is employed to get the best performance from the model. The rest of all models follow the same procedure where every model's performance comparison is done using the same technique. Initially, trials run the model for these epoch values and compare the training, validation loss and finally early stopping criteria is used. In connection with the mobile net model architecture, MobileNet is the lightweight model in deep neural network architecture as it consists of fewer parameters in the case of depth-wise layers. In our experiment, Mobile net showed its best performance with maximum accuracy of 74% being received using the early stopping criteria. It is worth visualizing the model performance to understand the model performance as shown in Figure 5, Figure 6, and Figure 7, and finally, Figure 8 shows the overall accuracy of the hybrid MobileNet model.

### Hybrid VGG16 Model Performance Comparision based

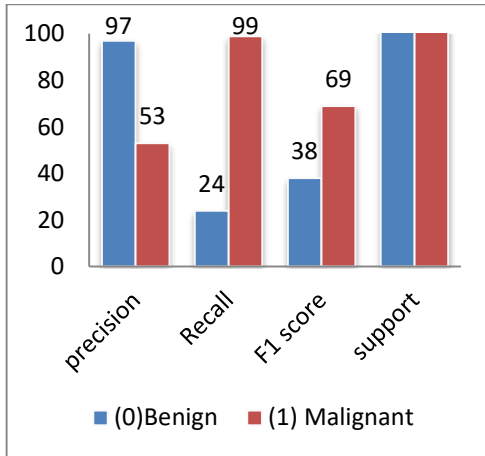


Fig. 9. Model Performance for epoch 100

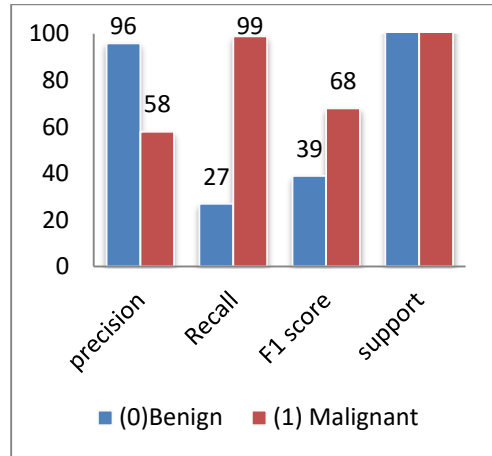


Fig. 10. Model performance for epoch 200

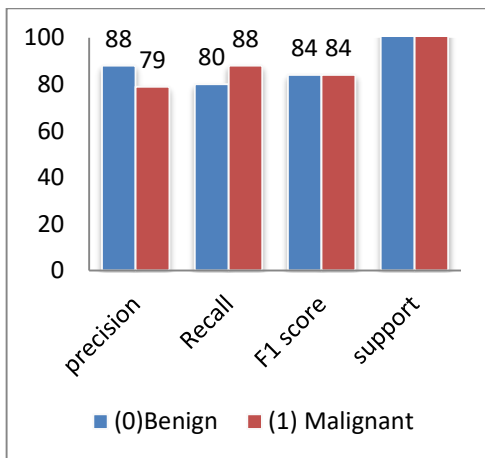


Fig. 11. Early stopping at epoch 43

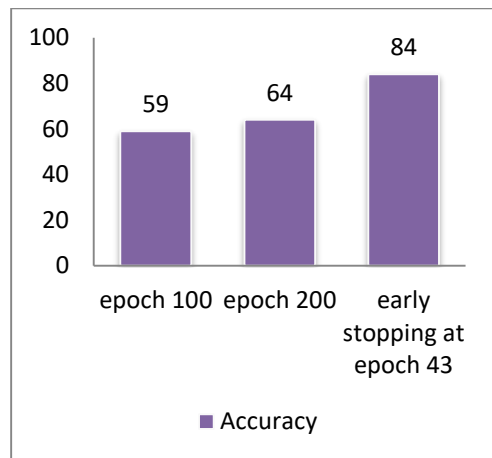


Fig. 12. Accuracy of the model

Results obtained for hybrid VGG16 model architecture using the same experimental criteria are discussed further. VGG16 model improves AlexNet by replacing large kernel-sized filters with multiple  $3 \times 3$  kernel-sized filters one after another. This is being the heavier model in terms of number of parameters, consumes more time during the training. The following graphs in figure 9, Figure 10, Figure 11, and Figure 12 visualize the overall results.

## Hybrid ResNet 101 Model Performance Comparison

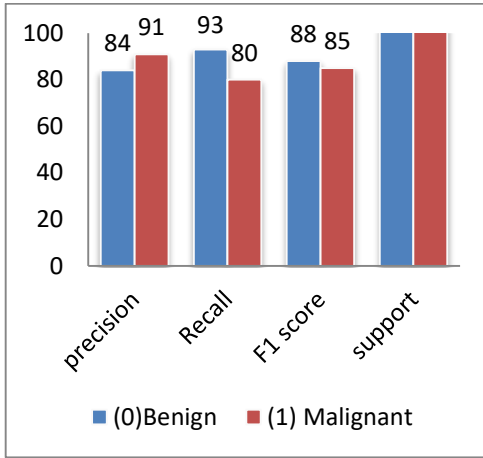


Fig. 13. Model Performance for epoch 100

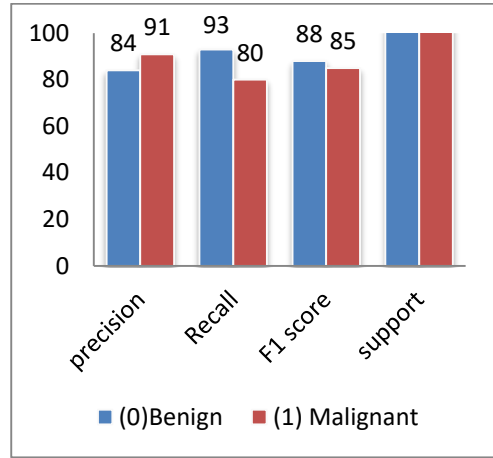


Fig. 14. Model performance for epoch 200

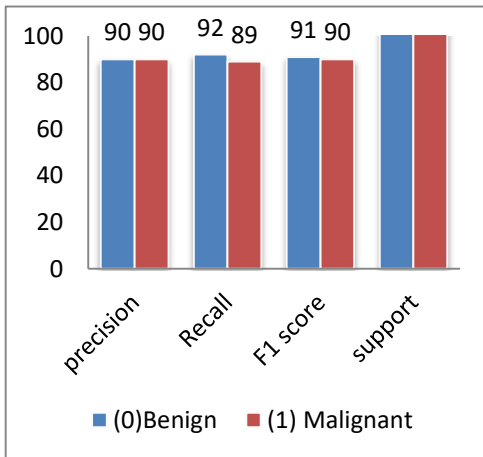


Fig. 15. Early stopping at epoch 103

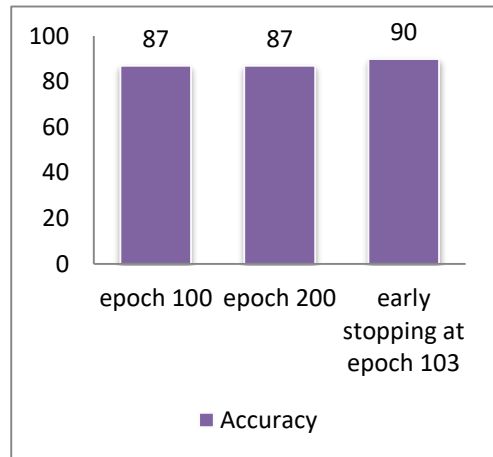
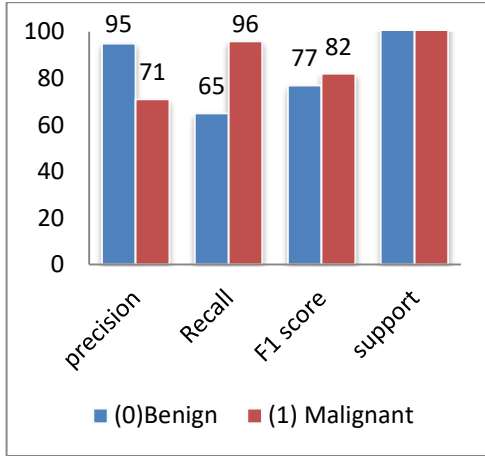


Fig. 16. Accuracy of the model

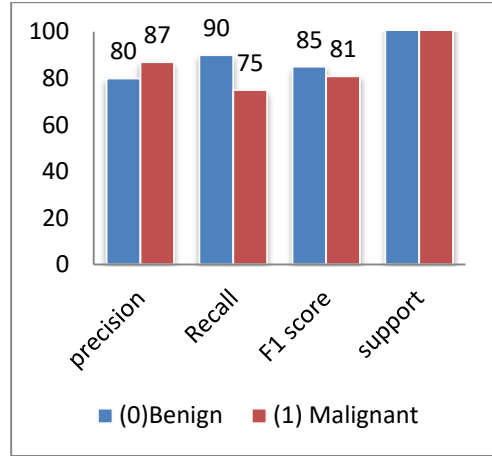
Figure 13, Figure 14, Figure 15, and Figure 16 compare the experimental comparison of ResNet101 model using different epoch values.

From these results, it is also noted that the ResNet 101 model shows highly favorable results intern showing its better suitability in classifying the histopathology images. It is worth noting that the number of parameters used by ResNet architecture is lighter compared to VGG16 architecture and hence it reduces the overhead of increased training time and computational complexity. Further work implemented the Hybrid ResNet 152 model with increased layers to demonstrate the model's capability to classify the given training data. Performance of these models is shown in the Figure 17, Figure 18, Figure 19 and Figure 20.

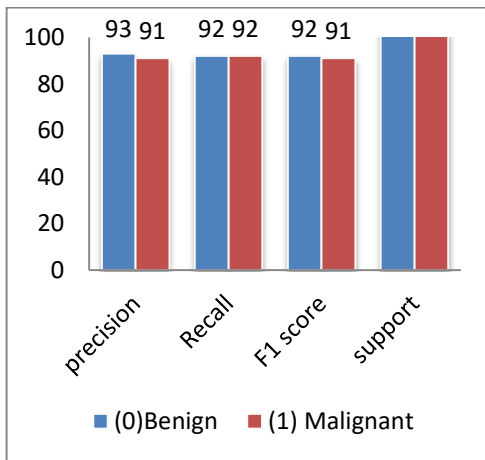
### Hybrid ResNet152 Model Performance Comparison



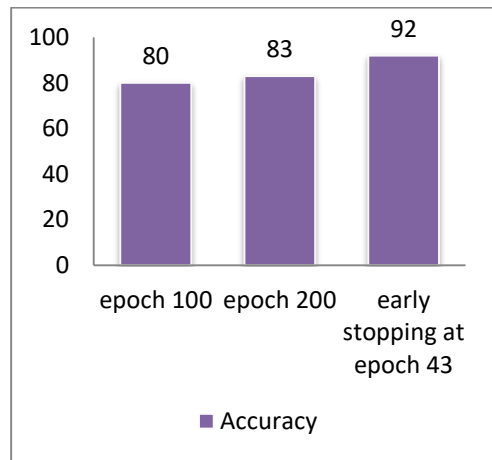
**Fig. 17. Model Performance for epoch 100**



**Fig. 18. Model performance for epoch 200**

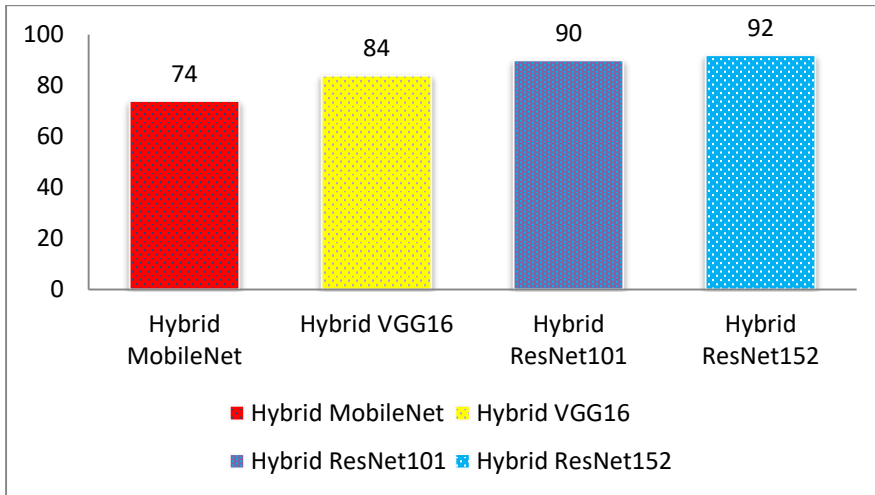


**Fig. 19. Early stopping at epoch 74**



**Fig. 20. Accuracy of the model**

It is evident from these results that the Hybrid ResNet 152 model is highly suitable for the classification of given data and adopts comparatively most suitable learning accuracy. Further, we claim that out of all proposed variants of models the Hybrid ResNet 152 model is well suited for the classification of histopathology cancer images. The following Figure 21 visualize the comparison of all models' accuracy in one chart.



**Fig. 21. Accuracy of Hybrid models**

Figure 21 summarize the overall accuracy of models, in which Hybrid ResNet152 has surpassed the other model with 92% of accuracy. Hence it is justified that developed hybrid deep learning architecture is highly suitable and capable of classifying the cancer images efficiently.

## 5. CONCLUSION

This study proposed Hybrid deep CNN-based architectures for the classification of histopathology cancer images. This study presents techniques to reuse the existing deep CNN architectures for Histopathology image classification. In order to improve the performance of existing architecture LSTM network used to retain and learn relationship between features across multiple layers. The proposed hybrid architectures use parallel LSTM network layers. Implemented models use TensorFlow-GPU combined with CUDA libraries to accelerate the training time required to train these complex architectures. Out of proposed Hybrid models, the Hybrid ResNet-based architectures with 101 and 152 layers have shown high accuracy of 90% and 92% and hence it is justified that these models are well suited for the given problem. The developed architectures can be extended with any histopathology image dataset using transfer learning technique, which remains one of the advantages in reusing these models.

## REFERENCES

- Aziz, H. A. (2017). A review of the role of public health informatics in healthcare. *Journal of Taibah University Medical Sciences*, 12(1), 78–81. <https://doi.org/10.1016/J.JTUMED.2016.08.011>
- Boumaraf, S., Liu, X., Zheng, Z., Ma, X., & Ferkous, C. (2021). A new transfer learning based approach to magnification dependent and independent classification of breast cancer in histopathological images. *Biomedical Signal Processing and Control*, 63, 102192. <https://doi.org/10.1016/j.bspc.2020.102192>
- Buddhavarapu, V. G., & Jothi, A. A. J. (2020). An experimental study on classification of thyroid histopathology images using transfer learning. *Pattern Recognition Letters*, 140, 1–9. <https://doi.org/10.1016/j.patrec.2020.09.020>

- Deep Learning Frameworks. NVIDIA Developer. (n.d.). Retrieved April 3, 2021 from <https://developer.nvidia.com/deep-learning-frameworks>
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Li, F.-F. (2010). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255). IEEE. <https://doi.org/10.1109/CVPR.2009.5206848>
- Djellali, C., Adda, M., & Moutacalli, M. T. (2020). A Data-Driven Deep Learning Model to Pattern Recognition for Medical Diagnosis, by using Model Aggregation and Model Selection. *Procedia Computer Science*, 177, 387–395. <https://doi.org/10.1016/J.PROCS.2020.10.052>
- Dwivedi, Y. K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., Duan, Y., Dwivedi, R., Edwards, J., Eirug, A., Galanos, V., Ilavarasan, P. V., Janssen, M., Jones, P., Kar, A. K., Kizgin, H., Kronemann, B., Lal, B., Lucini, B., Medaglia, R., Meunier-FitzHugh, K. L., Meunier-FitzHugh, L. C. L., Misra, S., Mogaji, E., Sharma, S. K., Singh, J. B., Raghavan, V., Raman, R., Rana, N. P., Samothrakis, S., Spencer, J., Tamilmani, K., Tubadji, A., Walton, P., & Williams, M. D. (2021). Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *International Journal of Information Management*, 57, 101994. <https://doi.org/10.1016/J.IJINFOMGT.2019.08.002>
- Eelbode, T., Sinouel, P., Maes, F., & Bisschops, R. (2021). Pitfalls in training and validation of deep learning systems. *Best Practice & Research Clinical Gastroenterology*, 52–53, 101712. <https://doi.org/10.1016/J.BPG.2020.101712>
- Guan, Q., Wang, Y., Ping, B., Li, D., Du, J., Qin, Y., Lu, H., Wan, X., & Xiang, J. (2019). Deep convolutional neural network VGG-16 model for differential diagnosing of papillary thyroid carcinomas in cytological images: a pilot study. *Journal of Cancer*, 10(20), 4876. <https://doi.org/10.7150/JCA.28769>
- Haghighat, E., & Juanes, R. (2020). ScienceDirect SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, 373, 113552. <https://doi.org/10.1016/j.cma.2020.113552>
- Improving the convergence of back-propagation learning with second-order methods — NYU Scholars. (n.d.). Retrieved March 23, 2022 from <https://nyuscholars.nyu.edu/en/publications/improving-the-convergence-of-back-propagation-learning-with-second-order-methods>
- Kaur, K., & Mittal, S. K. (2020). Classification of mammography image with CNN-RNN based semantic features and extra tree classifier approach using LSTM. *Materials Today: Proceedings*, in press. <https://doi.org/10.1016/j.matpr.2020.09.619>
- Kaur, P., Singh, G., & Kaur, P. (2019). Intellectual detection and validation of automated mammogram breast cancer images by multi-class SVM using deep learning classification. *Informatics in Medicine Unlocked*, 16, 100151. <https://doi.org/10.1016/J.IMU.2019.01.001>
- Leen, T. K., Dietterich, T. G., & Tresp, V. (2001). *Advances in Neural Information Processing Systems 13: Proceedings of the 2000 Conference*. MIT Press.
- Liang, R. B., Li, P., Li, B. T., Jin, J. T., Rusch, V. W., Jones, D. R., Wu, Y. L., Liu, Q., Yang, J., Yang, M. Z., Li, S., Long, H., Fu, J. H., Zhang, L. J., Lin, P., Rong, T. H., Hou, X., Lin, S. X., & Yang, H. X. (2021). Modification of Pathologic T Classification for Non-small Cell Lung Cancer With Visceral Pleural Invasion: Data From 1,055 Cases of Cancers  $\leq 3$  cm. *Chest*, 160(2), 754–764. <https://doi.org/10.1016/J.CHEST.2021.03.022>
- Moon, J. C. C., Perez De Arenaza, D., Elkington, A. G., Taneja, A. K., John, A. S., Wang, D., Janardhanan, R., Senior, R., Lahiri, A., Poole-Wilson, P. A., & Pennell, D. J. (2004). The Pathologic Basis of Q-Wave and Non-Q-Wave Myocardial Infarction: A Cardiovascular Magnetic Resonance Study. *Journal of the American College of Cardiology*, 44(3), 554–560. <https://doi.org/10.1016/J.JACC.2004.03.076>
- Pramanik, P. K. D., Pal, S., Mukhopadhyay, M., & Singh, S. P. (2021). Big Data classification: techniques and tools. *Applications of Big Data in Healthcare, 2021*, 1–43. <https://doi.org/10.1016/B978-0-12-820203-6.00002-3>
- Sarwinda, D., Paradisa, R., Bustamama, A., & Anggiab, P. (2021). Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer. *Procedia Computer Science*, 179, 423–431. <https://doi.org/10.1016/j.procs.2021.01.025>
- Sertolli, B., Ren, Z., Schuller, B. W., & Cummins, N. (2021). Representation transfer learning from deep end-to-end speech recognition networks for the classification of health states from speech. *Computer Speech and Language*, 68, 101204. <https://doi.org/10.1016/j.csl.2021.101204>
- Simonyan, K., & Zisserman, A. (2015). *Very deep convolutional networks for large-scale image recognition*. <http://www.robots.ox.ac.uk>

- Spanhol, F. A., Oliveira, L. S., Petitjean, C., & Heutte, L. (2016). A Dataset for Breast Cancer Histopathological Image Classification. *IEEE Transactions on Biomedical Engineering*, 63(7), 1455–1462. <https://doi.org/10.1109/TBME.2015.2496264>
- TensorFlow Framework & GPU Acceleration. NVIDIA Data Center. (n.d.). Retrieved March 23, 2022 from <https://www.nvidia.com/en-sg/data-center/gpu-accelerated-applications/tensorflow/>
- Tripathi, S., Singh, S. K., & Lee, H. K. (2021). An end-to-end breast tumour classification model using context-based patch modelling – A BiLSTM approach for image classification. *Computerized Medical Imaging and Graphics*, 87, 101838. <https://doi.org/10.1016/j.compmedimag.2020.101838>
- UCI Machine Learning Repository. (n.d.). Retrieved March 23, 2022 from <https://archive.ics.uci.edu/ml/index.php>
- Xiang, Q., Zhang, G., Wang, X., Lai, J., Li, R., & Hu, Q. (2019). Fruit image classification based on Mobilenetv2 with transfer learning technique. *ACM International Conference Proceeding Series* (pp. 1–7). Association for Computing Machinery. <https://doi.org/10.1145/3331453.3361658>
- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., & He, Q. (2021). A Comprehensive Survey on Transfer Learning. *Proceedings of the IEEE*, 109(1), 43–76. <https://doi.org/10.1109/JPROC.2020.3004555>



*Keywords: computer applications, robot safety, virtual reality*

*Jakub ANCZARSKI\**, *Adrian BOCHEN\**, *Marcin GŁĄB\**,  
*Mikołaj JACHOWICZ\**, *Jacek CABAN* [0000-0002-7546-8703]\*\*,  
*Radosław CECHOWICZ* [0000-0002-0074-1285]\*\*

## **A METHOD OF VERIFYING THE ROBOT'S TRAJECTORY FOR GOALS WITH A SHARED WORKSPACE**

### **Abstract**

*The latest market research (Fanuc Polska 2019) shows that the robotization of the Polish industry is accelerating. More and more companies are investing in robotic production lines, which enable greater efficiency of implemented processes and reduce labour costs. The article presents the possibilities of using virtual reality (VR) for behavioural analysis in open robotic systems with a shared workspace. The aim of the article is to develop a method of verification of programmed movements of an industrial robot in terms of safety and efficiency in systems with a shared workspace. The method of the robot program verification on the digital model of the working cell made in VR will be checked. The obtained research results indicate a great potential of this method in industrial applications as well as for educational purposes.*

### **1. INTRODUCTION**

Virtual reality (VR) is used by many spheres of human activity. Until recently, most people associated VR with the entertainment sphere, mainly for computer games, while for many years it has been used in scientific research (Klarak et al., 2021; Kuts, Otto, Tahemaa & Bondarenko, 2019; Sobaszek, Gola & Świć, 2022) and for educational purposes (Ji, Yin & Wang, 2018). For example, it can be used in architecture or design, where people can firstly try their projects in a digital environment before going to real projects (Kuts, Otto, Tahemaa & Bondarenko, 2019). This way can prevent dozens of mistakes and errors without any loss (Kuts, Otto, Tahemaa & Bondarenko, 2019). VR simulation could also be used in manufacturing, robotics and control systems (Chen et al., 2018; Covaciu et al., 2018; Kot, Novak & Bajak, 2018; Oyekan et al., 2019) by simulating different algorithms and control methods. Another application is audio – to use VR as a sound visualization tool for artists or the blind (Kose et al., 2018). Moreover, smart racks and simulations are done with force feedback (Burdea, 1999) for a better haptic feeling of tested in VR joysticks, devices or manipulators (Kuts, Otto, Tahemaa & Bondarenko, 2019). Fedorko (2021), investigated an application possibilities of the VR within the failure analysis of the rubber-textile

---

\* Lublin University of Technology, Faculty of Mechanical Engineering, Student, Poland  
Lublin University of Technology, Faculty of Mechanical Engineering, Department of Automation,  
Nadbystrzycka 36, 20-618 Lublin, Poland, j.caban@pollub.pl, r.cechowicz@pollub.pl

conveyor belts. Other studies concern the use of VR for robotic assembly works. Through VR proposed setup a typical robotic cell that performs assembly operations could rapidly be reprogramed to handle different assembly variations (Togias, Gkournelos, Angelakis, Michalos & Makris, 2021). The energetic and economic aspects of VR implantation in robotized technology systems are the subject of research Klačková et al. (Klačková, Kuric, Zajacko & Tucki, 2020). Economic aspects of manufacturing systems design e.g. capacity planning, system configuration and their effect on financial and operation costs are presented in (Gola, 2014).

The development of manufacturing processes is related to the growing quality and cost requirements of products (Bogucki, Stączek & Płaska, 2003; Cechowicz, 2003; Stączek, Bogucki & Płaska, 2003; Świć & Gola, 2013) as well as the development of automation and robotics systems. Automation is a process of replacing man's control function by operation of various machines and devices (Blatnický, Dižo & Timoščuk, 2016). The highly visible development of robotic systems is also noticeable in the automotive industry, as evidenced by the following scientific works (Blatnický, Dižo, Barta & Drożdźiel, 2020; Blatnický et al., 2020; Gola, Plinta & Grznar, 2021; Heydaryan, Suaza Bedolla & Belingardi, 2018; Jenis, Hrcek, Drumercik & Bastovansky, 2021) and aviation industry (Bistak et al., 2017; Vosniakos, Ouillon & Matsas, 2019). Handling with materials and products in the automotive industry is an activity, which requires the use of suitable operations methods for the transportation of a given quantity of material to a specified location in as short a time as possible (Blatnický et al., 2020). Vosniakos et al. (Vosniakos, Ouillon & Matsas, 2019) analyzes the application of VR to the safety strategy of robots cooperating in the technological line of composite materials. In (Wang, Chen, Jiao, Johnson & Zhang, 2019) a human-robot collaborative welding system in virtual reality was proposed, which allows them to collaborate with each other in order to carry out welding tasks. The welding experiments show that the welded workpiece from human-robot collaboration has better performance compared with that from either humans or robots separately and demonstrated the effectiveness of the proposed virtual reality human-robot collaborative welding system (Wang, Chen, Jiao, Johnson & Zhang, 2019). The VR technique is already widely used to analyze robotic systems and verify the technological correctness of programs (Ehmanna & Wittenberg, 2018; Heydaryan, Suaza Bedolla & Belingardi, 2018; Kuts, Otto, Tahemaa & Bondarenko, 2019; Shen, 2020).

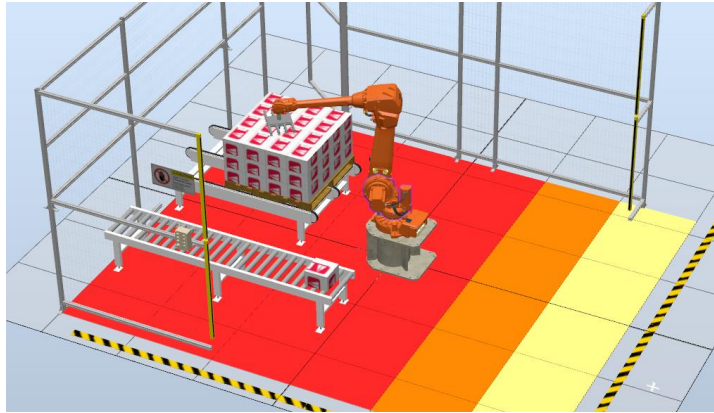
The latest market research (Wpływ robotyzacji na konkurencyjność polskich przedsiębiorstw III edycja, 2019) shows that the robotisation of the Polish industry is accelerating. There are new security systems on the market that allow industrial robots to work in open spaces (without fencing) and to share space between a machine and a human (e.g. FreeMove, – Veo Robotics, 2020; Collaborative Robot Safety Made Simple, 2020). Preparing an industrial robot to work in such conditions requires taking into account additional requirements (e.g. traffic predictability, traffic range, relation to the shared area) and carrying out an extended safety analysis. It is also important to program the robot in such a way that the presence of a human in the shared zone does not cause an excessive decrease in system efficiency.

The main aim of the article is to develop a method to verify the programmed movements of an industrial robot in terms of safety and performance in systems with a shared workspace. The method of verifying the robot program on a digital model of the working cell made in VR will be checked. We propose to expand the field of application of this method with behavioural analysis in open robotic systems with a shared workspace.

## 2. RESEARCH METHODOLOGY

The simulation tests were carried out in the ABB RobotStudio offline simulation and programming environment. VR goggles were used to verify the complete installation of the robot's trajectory in a virtual environment.

The verification of the robot path and safety features of a robotic cell, a digital model was created. A simple palletizing cell (Fig. 1) was chosen for the experiment. The access to the cell was open at one side. The open side of the cell was protected by a simulated safety system that would react (decrease the speed of the robot or stop the robot completely) if a presence of a human operator or a transport vehicle was detected.



**Fig. 1. Robot cell used in the experiment**

The cell was equipped with a SSM (Speed and Safety Monitoring) safety system. The dimensions of the safety zones (marked with red, orange and yellow) were calculated according to the requirements specified in ISO/TS 15066 and discussed in (Szabo et al. 2012). The required separation distance  $S$  can be obtained from formula:

$$S = K_H (T_R + T_B) + K_R (T_R) + B + C \quad (1)$$

where:  $K_H$  is the approach speed of human operator – assumed 1.6 m/s as suggested in TS15066,  $K_R$  is the maximum robot speed obtained from kinematic calculations (full speed: 3 m/s, reduced speeds: 1.5 m/s in with operator in the yellow zone and 0.3 m/s with operator in the orange zone),  $T_R$  – system reaction time (assumed 0.21 s),  $T_B$  – time to full stop (braking time; obtained from simulation),  $B$  – stop distance (obtained from simulation),  $C$  – minimum separation distance (assumed 1.08 m after ISO 13855).

The calculated separation distances for each zone are presented in table 1. The zones used in the model (Fig. 1) are slightly oversized for clarity.

The main purpose of the experiment was to verify the safety of the human operator during ad-hoc maintenance actions:

- making adjustments to the box on the infeder,
- adjusting the position of the box in the pallet,
- stopping the robot at the picking position.

Furthermore, the reaction of the system to short-term safety zone intrusion (like people passing) was tested.

**Tab. 1. Stop times and separation distances obtained from calculations**

Area	Outside yellow zone	Yellow zone	Orange zone
<b>Robot tool speed</b>	3 m/s	1.5 m/s	0.3 m/s
<b>Robot stop time (<math>T_B</math>)</b>	0.5 s	0.2 s	0.04 s
<b>Robot stop distance (including the reaction time delay)</b>	0.95 m	0.4 m	0.06 m
<b>Separation distance (<math>S</math>)</b>	2.8 m	2.0 m	1.5 m

### 3. RESULTS OF THE EXPERIMENT

This chapter presents the results of the experiment obtained from the simulation of the states of human-robot cooperation depending on the situation in the palletizing cell. The area in front of the robot is divided into 3 zones, after the operator enters the restricted area, the robot slows down (first to 50% in the first, yellow zone, then to 10% in the second, orange zone). After the employee enters the last, third zone, the robot stops all movements.

Fig. 2 shows the situation in which the robot works in the presence of a human being in the second safety zone. The safety zones in the robot cell are color-coded as follows: 1<sup>st</sup> zone – yellow, 2<sup>nd</sup> zone – orange, 3<sup>rd</sup> zone – red.



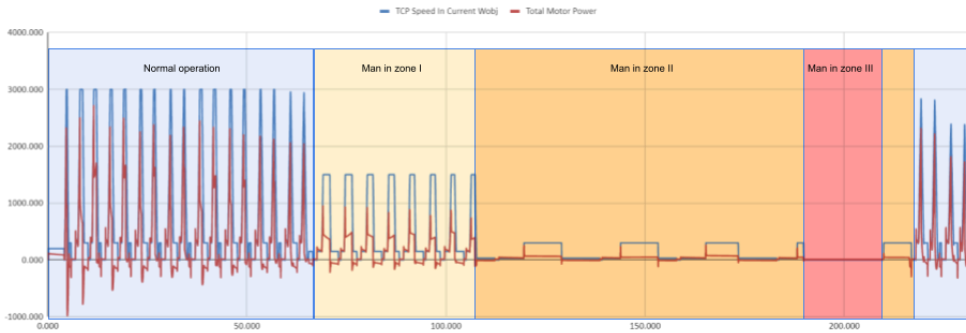
**Fig. 2. Human operator in the 2<sup>nd</sup> safety zone**

The power of robot motors and the resulting TCP (tool) speed are shown in Fig. 3. Limiting the speed of the robot helps to accomplish two objectives:

- 1) to decrease the distance at which the robot stops to safe values,
- 2) to reduce the stress level (or increase psychological comfort) of the person approaching the robot.

To find out how people would react, some volunteers were requested to enter the robot cell and perform one of the maintenance actions in a virtual reality setting (like verifying the

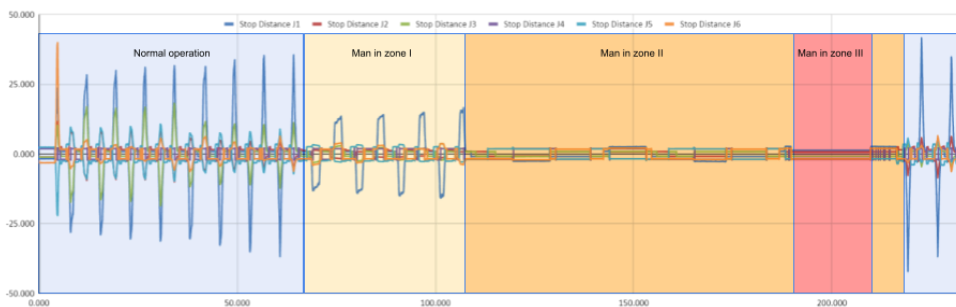
sealing on of the box in the in-feed conveyor). The tests have shown that people felt greater comfort and confidence when they saw robots reacting to their actions by slowing down and then stopping completely. When the safety system was disengaged and the robot did not react to the presence of humans, they had shown stressful reactions and tried to escape from the cell quickly.



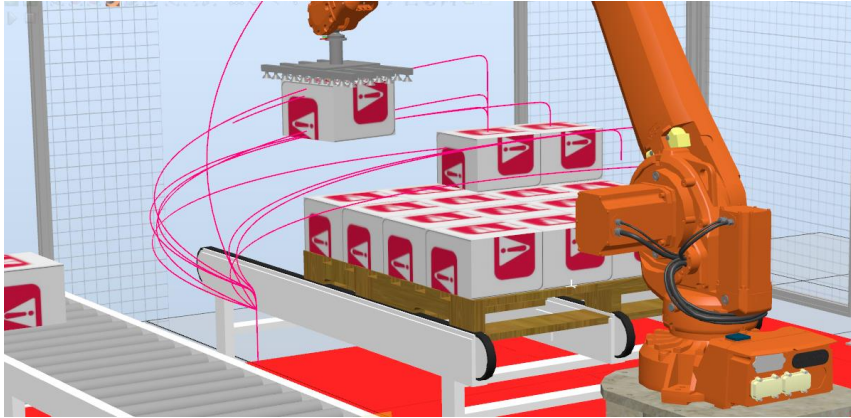
**Fig. 3. Signals recorded during simulation: speed of manipulator and motor power – colours of the zones in the graph correspond with the safety zones defined in the robot cell**

The approximated category 1 (normal stop) stop distances were computed for each robot axis using the functions built into the RobotStudio and are shown in Fig. 4. The computations confirmed the feelings of the volunteers, that the robot speed in the second (orange) zone allowed them to stop the robot almost immediately (computed distance is less than 10 mm in each axis). The volunteers felt “at control” of the situation and felt confident to approach the box on the conveyor or the pallet. The experiment has shown that volunteers quickly gained confidence in the requested tasks and soon were performing them almost without any consideration of the actual robot position and speed. This overconfidence could create dangerous situations in the real world and should be addressed in further research. These findings show, that all the personnel working near the open robot cells should at least have appropriate training and that the safety procedures should be strictly imposed in such environments.

The experiment has also shown that the recovery time of the open cell was shorter than the estimated re-starting time of a traditional, fenced cell. This in normal working conditions, that is assuming no intrusions in the safety zones except for the process servicing, would result in greater productivity of the cell.



**Fig. 4. Signals recorded during simulation concern the stop distance – colours of the zones in the graph correspond with the safety zones defined in the robot cell**



**Fig. 5. Determining the trajectory of the robot's work on the palletizing station**

Virtual simulation environment also provided a powerful tool for analysing and optimisation of robot trajectories (Fig. 5). The programmers with little robot experience could make better decisions when they saw the robot path in 3D. This applied especially to assessing the approach and depart distances and to the choice of joint or linear (circular) movements.

#### **4. CONCLUSIONS**

The article describes the design and implementation example of a VR-based system to provide the user with a good understanding of the current production environment, as well as the possibility of reprogramming the existing robotic cell. The benefits of using VR consist in increasing the flexibility of the currently installed robotic cell in a cost-effective manner and based on minimal hardware changes.

On the basis of the conducted analyses, a clear division into safety zones can be noticed, depending on the presence of a human-operator in the robot cell. There are clearly noticeable drops in the value of the manipulator speed and the engine power values in individual safety zones. Conducted simulations with volunteers using VR goggles show human behaviour in a robot cell with a shared work space. The main conclusion is that a person quickly gets used to working in such a system, and seeing that the robot changes its operating parameters depending on the place occupied by the operator, it gains confidence. Overconfidence can create dangerous situations in the real world and should be the subject of further research.

Thanks to the use of the VR technique, it is possible to obtain better results of robot programming and reduce errors in the actual adjustment of the cell with an industrial robot to work in a shared area. In addition, the VR technique is a very good educational tool that allows young engineers and operators of industrial machines, including robots and internal transport devices to understand the processes taking place in the production space more easily.

## Funding

*The project was financed in the framework of the project Lublin University of Technology – Regional Excellence Initiative, funded by the Polish Ministry of Science and Higher Education (contract no. 030/RID/2018/19).*

## Conflicts of Interest

*The authors declare no conflict of interest.*

## REFERENCES

- Bistak, M., Medvecký, S., Gajdosova, E., Dzimko, M., Gramblicka, S., Kohar, R., Stopka, M., Steininger, J., Hrcak, S., Tropp, M., & Brumerčík, F. (2017). Applications of modern technologies in the production of aircraft propeller prototype. *Communications - Scientific Letters of the University of Zilina*, 19(2), 54–59. <https://doi.org/10.26552/com.C.2017.2A.54-59>
- Blatnický, M., Dižo, J., Barta, D., & Drożdźiel, P. (2020). FEM analysis of main parts of a manipulator for mounting a compressor to a car equipped with a pneumatic suspension system. *Diagnostyka*, 21(2), 87–94. <https://doi.org/10.29354/diag/122549>
- Blatnický, M., Dižo, J., Gerlici, J., Sága, M., Lack, T., & Kuba, E. (2020). Design of a robotic manipulator for handling products of automotive industry. *International Journal of Advanced Robotic Systems*, 17(1), 1–11. <https://doi.org/10.1177/1729881420906290>
- Blatnický, M., Dižo, J., & Timoščuk, M. (2016). Design of a three-finger robot manipulator. *Manufacturing Technology*, 16(3), 485–489.
- Bogucki, M., Stączek, P., & Płaska, S. (2003). Methods of improving quality product and process using experimental techniques. *Second International CAMT Conference (Centre for Advanced Manufacturing Technologies), Modern Trends in Manufacturing* (pp. 15–20).
- Burdea, G. C. (1999). Invited review: the synergy between virtual reality and robotics. *IEEE Transactions on Robotics and Automation*, 15(3), 400–410. <https://doi.org/10.1109/70.768174>.
- Cechowicz, R. (2003). An approach to flexible scheduling in job shop manufacturing system. *Second International CAMT Conference (Centre for Advanced Manufacturing Technologies), Modern Trends in Manufacturing* (pp. 27–35).
- Chen, C., Su, B., Guo, M., Zhong, Y., Yang, Y., & Kuo, H. L. (2018). Applying virtual reality to control of logical control mechanism system. *IEEE International Conference on Applied System Invention (ICASI)* (pp. 520–523). IEEE. <https://doi.org/10.1109/ICASI.2018.8394302>
- Collaborative Robot Safety Made Simple. (2020). <https://sickusablog.com/collaborative-robot-safety-made-simple>
- Covaci, F., Pislă, A., Carbone, G., Puskas, F., Vaida, C., & Pislă, D. (2018). VR interface for cooperative robots applied in dynamic environments. *IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)* (pp. 1–6). IEEE. <https://doi.org/10.1109/AQTR.2018.8402734>
- Ehmanna, D., & Wittenberg, C. (2018). The idea of Virtual Teach-In in the field of industrial robotics. *2018 IEEE 14th International Conference on Control and Automation (ICCA)* (pp. 680–685). IEEE. <http://dx.doi.org/10.1109/ICCA.2018.8444250>
- Fedorko, G. (2021). Application possibilities of virtual reality in failure analysis of conveyor belts. *Engineering Failure Analysis*, 128, 105615. <https://doi.org/10.1016/j.engfailanal.2021.105615>
- FreeMove — Veo Robotics. (2020). <https://www.veobot.com/freemove>
- Gola, A. (2014). Economic aspects of manufacturing systems design. *Actual Problems of Economics*, 156(6), 205–212.
- Gola, A., Plinta, D., & Grzñar, P. (2021). Modelling and simulation of reconfigurable manufacturing system for machining of casing-class parts. *Engineering for Rural Development*, 20, 1563–1568.



- Heydaryan, S., Suaza Bedolla, J., & Belingardi, G. (2018). Safety Design and Development of a Human-Robot Collaboration Assembly Process in the Automotive Industry. *Applied Sciences*, 8(3), 344. <https://doi.org/10.3390/app8030344>
- Ji, W., Yin, S., & Wang, L. (2018). A virtual training based programming-free automatic assembly approach for future industry. *IEEE Access*, 6, 43865–43873. <https://doi.org/10.1109/ACCESS.2018.2863697>
- Jenis, J., Hrcek, S., Brumerick, F., & Bastovansky, R. (2021). Design of automatic assembly station for industrial vehicles parts. *LOGI – Scientific Journal on Transport and Logistics*, 12, 1, 204–213. <https://doi.org/10.2478/logi-2021-0019>
- Klačková, I., Kuric, I., Zajacko, I., & Tucki, K. (2020). Energy and economical aspects of implementation of virtual reality in robotized technology systems. *ICETA 2020 – 18<sup>th</sup> IEEE International Conference on Emerging eLearning Technologies and Applications, Proceedings* (pp. 318–322). IEEE. <https://doi.org/10.1109/ICETA51985.2020.9379176>
- Klarak, J., Kuric, I., Cisar, M., Stanček, J., Hajducik, A., & Tucki, K. (2021). Processing 3D data from laser sensor into visual content using pattern recognition. *2021 IEEE 8<sup>th</sup> International Conference on Industrial Engineering and Applications (ICIEA)* (pp. 543–549). IEEE. <https://doi.org/10.1109/ICIEA52957.2021.9436712>
- Kose, A., Tepljakov, A., Astapov, S., Draheim, D., Petlenkov, E. K., & Vassiljeva, K. (2018). Towards a synesthesia laboratory: real-time localization and visualization of a sound source for Virtual Reality applications. *Journal of Communications Software and Systems*, 14(1), 112–120. <http://dx.doi.org/10.24138/jcomss.v14i1.410>
- Kot, T., Novák, P., & Bajak, J. (2018). Using HoloLens to Create a Virtual Operator Station for Mobile Robots. *19<sup>th</sup> International Carpathian Control Conference (ICCC)* (pp. 422–427). IEEE. <https://doi.org/10.1109/CarpathianCC.2018.8399667>
- Kuts, V., Otto, T., Tähemaa, T., & Bondarenko, Y. (2019). Digital Twin based synchronised control and simulation of the industrial robotic cell using Virtual Reality. *Journal of Machine Engineering*, 19(1), 128–144. <https://doi.org/10.5604/01.3001.0013.0464>
- Oyekan, J. O., Hutabarat, W., Tiwari, A., Grech, R., Aung, M. H., Mariani, M. P., López-Dávalos, L., Ricaud, T., Singh, S., & Dupuis, C. (2019). The effectiveness of virtual environments in developing collaborative strategies between industrial robots and humans. *Robotics and Computer-Integrated Manufacturing*, 55, 41–54. <https://doi.org/10.1016/j.rcim.2018.07.006>
- Shen, W. (2020). Research on virtual simulation design of ABB robot welding operation based on Robotstudio. *IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)* (pp. 894–897). IEEE. <http://dx.doi.org/10.1109/ICAICA50127.2020.9182551>
- Sobaszek, Ł., Gola, A., & Świć, A. (2022). The algorithms for robust scheduling of production jobs under machine failure and variable technological operation times. *Lecture Notes in Mechanical Engineering*, (pp. 56–67). Springer. [https://doi.org/10.1007/978-3-030-78170-5\\_6](https://doi.org/10.1007/978-3-030-78170-5_6)
- Stączek, P., Bogucki, M., & Płaska, S. (2003). Fuzzy logic in supervising of complex technological processes. Second International CAMT Conference (Centre for Advanced Manufacturing Technologies), Modern Trends in Manufacturing (pp. 351–360).
- Świć, A., & Gola, A. (2013). Economic analysis of casing parts production in a flexible manufacturing system. *Actual Problems of Economics*, 141(3), 526–533.
- Szabo, S., Shackleford, W., Norcross, R., & Marvel, J. (2012). A testbed for evaluation of speed and separation monitoring in a human robot collaborative environment. NIST Interagency/Internal Report (NISTIR), National Institute of Standards and Technology, Gaithersburg, MD, <https://doi.org/10.6028/NIST.IR.7851>
- Togias, T., Gkourmelos, C., Angelakis, P., Michalos, G., & Makris, S. (2021). Virtual reality environment for industrial robot control and path design. *Procedia CIRP*, 100, 133–138. <https://doi.org/10.1016/j.procir.2021.05.021>
- Vosniakos, G. C., Ouillon, L., & Matsas, E. (2019). Exploration of two safety strategies in human-robot collaborative manufacturing using Virtual Reality. *Procedia Manufacturing*, 38, 524–531. <https://doi.org/10.1016/j.promfg.2020.01.066>
- Wang, Q., Cheng, Y., Jiao, W., Johnson, M. T., & Zhang, Y. M. (2019). Virtual reality human-robot collaborative welding: a case study of weaving gas tungsten arc welding. *Journal of Manufacturing Processes*, 48, 210–217. <https://doi.org/10.1016/j.jmapro.2019.10.016>
- Wpływ robotyzacji na konkurencyjność polskich przedsiębiorstw III edycja. (2019). *Instytut Prognoz i Analiz Gospodarczych*. Fanuc Polska Sp z o.o.



*Keywords: classification, remote sensing, soil adjusted vegetation index, normalized difference vegetation index, vegetation*

Anusha NALLAPAREDDY [0000-0002-2115-5998]\*

## DETECTION AND CLASSIFICATION OF VEGETATION AREAS FROM RED AND NEAR INFRARED BANDS OF LANDSAT-8 OPTICAL SATELLITE IMAGE

### Abstract

*Detection and classification of vegetation is a crucial technical task in the management of natural resources since vegetation serves as a foundation for all living things and has a significant impact on climate change such as impacting terrestrial carbon dioxide (CO<sub>2</sub>). Traditional approaches for acquiring vegetation covers such as field surveys, map interpretation, collateral and data analysis are ineffective as they are time consuming and expensive. In this paper vegetation regions are automatically detected by applying simple but effective vegetation indices Normalized Difference Vegetation Index (NDVI) and Soil Adjusted Vegetation Index (SAVI) on red(R) and near infrared (NIR) bands of Landsat-8 satellite image. Remote sensing technology makes it possible to analyze vegetation cover across wide areas in a cost-effective manner. Using remotely sensed images, the mapping of vegetation requires a number of factors, techniques, and methodologies. The rapid improvement of remote sensing technologies broadens possibilities for image sources making remotely sensed images more accessible. The dataset used in this paper is the R and NIR bands of Level-1 Tier 1 Landsat-8 optical remote sensing image acquired on 6<sup>th</sup> September 2013, is processed and made available to users on 2<sup>nd</sup> May 2017. The pre-processing involving sub-setting operation is performed using the ERDAS Imagine tool on R and NIR bands of Landsat-8 image. The NDVI and SAVI are utilized to extract vegetation features automatically by using python language. Finally by establishing a threshold, vegetation cover of the research area is detected and then classified.*

### 1. INTRODUCTION

Vegetation is a huge and complex subject spanning a wide range of plant life as shown in Fig. 1. There are many different classification systems for vegetation, each with its own set of criteria (Persson, Ulander & Soja, 2018, Ruiz et al., 2021). It has become a critical technological challenge for managing natural resources, and vegetation plays a significant role in global climate change such as impacting terrestrial CO<sub>2</sub> (Yu, Xie & Sha, 2008).

---

\* Vidya Jyothi Institute of Technology, Department of Computer Science and Engineering, India, anusha.nallapareddy@gmail.com

Vegetation extraction involves understating the Color, texture, tone, pattern, and association information in the satellite images (Gandhi, Parthiban & Christy, 2015). To do this, various methods are developed. These are often categorized into supervised and unsupervised methods (Porikli et al., 2017). Vegetation detection includes image pre-processing and image classification.



**Fig. 1. Vegetation cover**

### **1.1. Image Pre-processing**

Satellite imaging is used for a variety of purposes including assessing and monitoring the state of the earth's surface, disaster management and the study of natural disaster images, Land Use Land Cover (LULC) mapping, and LULC-based mapping modifications (Dalponte et al., 2020).

The goal of pre-processing operations is enhancing images (Bouhennache, Bouden & Taleb, 2014). To eliminate noise and improve the interpretability of image data, satellite images must be pre-processed before vegetation extraction (Yu, Xie & Sha, 2008). Pre-processing of remote sensing data includes geometric correction, atmospheric correction, and topographic correction (Jing et al., 2009). To account for surface reflectivity, all photos are atmospherically corrected (Fatihaa et al., 2013). The LEDAPS system is used to process the Landsat 5 and 7 images. Landsat-8 satellite images are processed by the Landsat-8 Surface Reflectance (L8SR) system (Schmidt et al., 2013).

Prior to vegetation extraction, satellite image preparation is required to eliminate noise and improve the interpretability of satellite image data. This is especially true when a temporal series of images is employed or when a large area is covered by multiple photographs, because it is critical to make these images spatially and spectrally consistent. Although image pre-processing varies depending on the sensor used, it usually entails a series of steps including radiometric correction, geometric correction, image enhancement, and masking (for clouds, water etc.,) (Yu, Xie & Sha, 2008).

The practice of fixing radiometric defects or distortions in images to enhance the accuracy of brightness values is known as radiometric correction of remote sensing data. Seasonal phenology, ground conditions, and atmospheric variables significantly influence multi-temporal spectral response variability. To identify noise, radiometric correction is essential when the spectral signals are insufficiently strong to exclude the impacts of these intricate components (Chen, Giri & Vogelmann, 2016).

Geometric correction's main purpose is to remove geometric distortions from affected images. It is done by creating a link between the image coordinate system and the geographic coordinate system utilizing the sensor's calibration data, position and altitude readings, and ground control points. As a result, geometric corrections require choosing a map projection system and co-registering satellite image data with other data that serves as a calibration reference. It is done by using the sensor's calibration data, position and altitude readings, and ground control points to create a link between the image coordinate system and the geographic coordinate system (Jing et al., 2009).

The USGS Earth Resources Observation and Science Centre provides Landsat-8 satellite products that are/can be radiometrically corrected, atmospherically corrected, terrain corrected (Li et al., 2017).

## **1.2. Vegetation Detection**

Among vegetation indices, NDVI is one of the most widely used techniques. It is the best indicator for crop condition and spatial distribution. A combination of red and near-infrared bands are used to determine NDVI (Jing et al., 2009).

NDVI is a good indicator for reflecting dynamic changes in vegetation groupings. Individual vegetation groups can be defined based on their distinct phenology or dynamic NDVI signals (Yu, Xie & Sha, 2008).

Huete proposed a SAVI index with an adjustment parameter, indicating  $L$  that characterizes the ground and its rate of vegetation cover. The isolignes vegetation is not aligned to the right-hand side of the grounds, but they cut this one in a place shown by Huete based on the density of vegetation cover. For a heavy density of vegetation, the parameter  $L$  is set to 0.25, while for a very low density of vegetation, it is set to 1. The value is equivalent to 0.5 at intermediate densities (Fatihaa et al., 2013).

## **1.3. Image Classification**

The technique of deriving unique classifications or themes (e.g., land use categories, vegetation species) from raw remotely sensed satellite data is known as image classification. Obviously, image pre-processing is included in this definition. The technique that follows image pre-processing is simply referred to as image classification. Traditional and improved methods are two techniques for extracting vegetation from pre-processed images (Yu, Xie & Sha, 2008).

Landsat-8 data is compared to older Landsat data, the surface reflectance of the six optical bands is measured, and two VIs are created (NDVI and EVI) and land use inventory map with labelled training samples and simple linear trend is used to detect changes and classify them as abrupt or gradual changes (Zhu et al., 2016).

Landsat images (He et al., 2020; Dutta, Rahman & Kundu, 2015) are used and sub pixel analysis is performed because each pixel may not belong to a single class (pure pixel). For dimensionality, Principal Component Analysis is utilized. Reduction and Linear Spectral unmixing is used for classification and simple image differencing is used to detect change (Dutta, Rahman & Kundu, 2015).

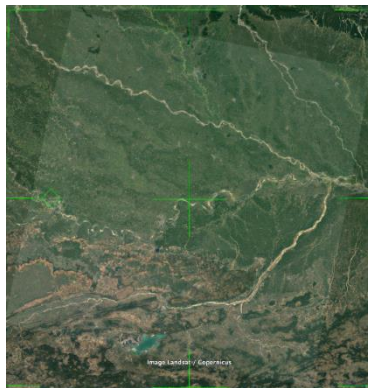
The CCDC algorithm is used to detect and classify land cover changes. Correction of the BRDF impact for each Landsat image in order to provide a more uniform LTS, with the hope of improving both change detection and classification. The Support Vector Machine (SVM) is a nonlinear classification approach that is used to classify images and forecast disease (He et al., 2020).

The spectra of the same type of surface object are approximately linearly scaled replicas of one another due to atmospheric and topographic influences. Spectral angle classifiers (SAC) both supervised and unsupervised are employed to account for this. The distances between pairs of signatures for categorization, as well as the classification of biotic communities and land cover, are successfully determined using these SACs (Yu, Xie & Sha, 2008).

## 2. METHODOLOGY

### 2.1. Study Area and Dataset Used

The study is conducted in districts of Uttar Pradesh with latitude  $24^{\circ}57'12.07''$ – $27^{\circ}01'54.86''$  North and longitude  $82^{\circ}14'49.38''$ – $84^{\circ}27'49.00''$  East, as this area has a large cover of vegetation. Fig. 2 shows a Google Earth Engine view of the research area.



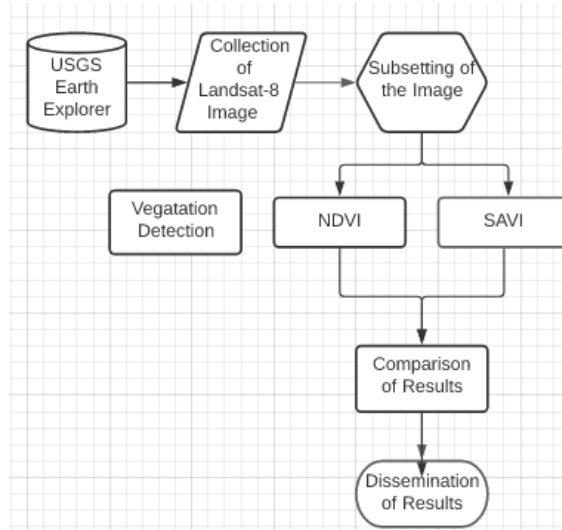
**Fig. 1. View of study area (Credits: Google Earth Engine platform)**

In this research, the satellite image collected by Landsat-8 is used. The USGS and NASA collaborate on the Landsat series to observe the earth. The Landsat-8 satellite image has a resolution of 30 meters. R and NIR bands are used to classify healthy vegetation. R and NIR bands have wavelengths of 0.64–0.67 and 0.85–0.88 micrometers respectively (Roy et al., 2014).

A Level-1 Tier 1 Landsat-8 satellite image is collected from the website USGS Earth Explorer, which provides free access to satellite images of different satellites. The satellite image collected contains information captured by both Landsat-8's operational land imager (OLI) and thermal infrared sensor (TIRS) sensors of 142 Worldwide Reference System (WRS) path and 042 WRS row. The image is Precision and Terrain Corrected (L1TP) (Landsat Missions, n.d.) and was acquired on 6<sup>th</sup> September 2013, is processed and made available to users on 2<sup>nd</sup> May 2017.

## 2.2. Research Methodology

Landsat-8 satellite image acquired on 6<sup>th</sup> September 2013 and made available to users on 2<sup>nd</sup> May 2017 is considered as the input. R and NIR bands of this image are used in this study to classify healthy vegetation. As the satellite image acquired is too large to be processed, sub-setting is performed on R and NIR bands of the input image to obtain an Area of Interest (AOI). The pre-processing step is carried out in ERDAS Imagine tool (Asokan et al., 2020; Abburu & Golla, 2015).



**Fig. 2. Flowchart of the applied methodology**

To extract the vegetation features (Ahmad et al., 2020; Langendoen et al., 2020) automatically, vegetation indices like NDVI, and SAVI are utilized (Sowmya, Deepa & Venugopal, 2017). Fig. 3 depicts the flow diagram of the applied methodology in this work. The vegetation features for the input satellite image are classified by applying threshold values for the computed vegetation indices (Xue & Su, 2017).

### 2.2.1. NDVI

NDVI is calculated as a ratio between R and NIR bands (Yu, Xie & Sha, 2008) using the Equation (1).

$$NDVI = \frac{NIR - R}{NIR + R} \quad (1)$$

In the image collected by the Landsat-8 satellite, band 5 corresponds to NIR band and band 4 corresponds to R band (Omar & Kawamukai, 2021; Rhyma et al., 2019). By substituting band 5 and band 4 in place of NIR and R bands respectively in Equation (1), NDVI for Landsat-8 image is calculated using the Equation (2).

$$NDVI = \frac{\text{band 5} - \text{band 4}}{\text{band 5} + \text{band 4}} \quad (2)$$

### 2.2.2. SAVI

SAVI is calculated as a ratio of the R and NIR bands with a soil brightness correction factor (L) (Rhyma et al., 2019) as given in Equation (3).

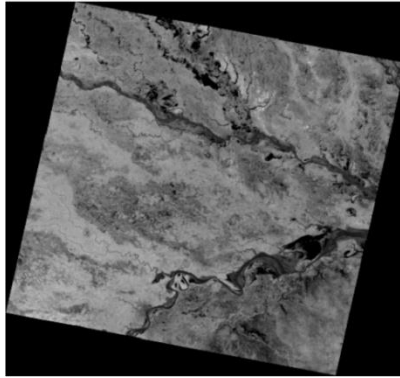
$$SAVI = (NIR - R) / (NIR + R) (1 + L) \quad (3)$$

By substituting band 5 and band 4 in place of NIR and R bands respectively in Equation (3) and L with 0.5 (Rhyma et al., 2019), SAVI for Landsat-8 image is calculated using the Equation (4).

$$SAVI = (band\ 5 - band\ 4) / (band\ 5 + band\ 4) (1 + L) \quad (4)$$

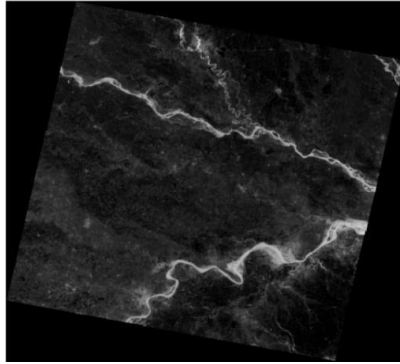
### 2.3. Technologies Used

The satellite image acquired is in GeoTiff format because it contains georeferencing information embedded in the GeoTiff file. Python 3.6.9 has a Rasterio library which can be used to work with this kind of input images. EarthPy is a python package devoted to working with spatial and remote sensing data. EarthPy has several Python package dependencies including rasterio, geopandas, numpy.



**Fig. 3. Subset of the R band of the Landsat-8 satellite image acquired 6<sup>th</sup> September 2013**

The input images are loaded into google drive and mounted on google collab, an online version of Anaconda jupyter. Fig. 4 and Fig. 5 depicts the red and near infrared bands respectively of Landsat-8 satellite acquired on 6<sup>th</sup> September 2013.

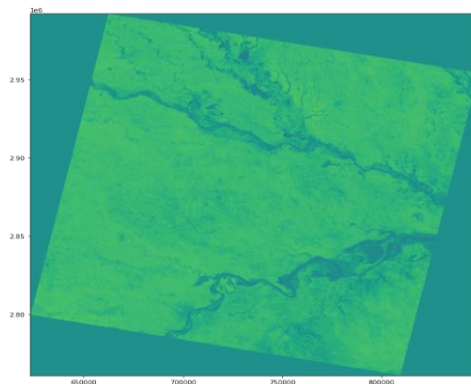


**Fig. 4. Subset of the NIR band of the Landsat-8 satellite image acquired on 6<sup>th</sup> September 2013**

### **3. RESULTS AND DISCUSSIONS**

A Landsat-8 satellite image is collected and pre-processing operation such as sub-setting is performed using ERDAS Imagine tool. The spectral bands required for calculating vegetation indices are opened and read using the rasterio library of python to perform the operations on the input image bands. The computed NDVI and SAVI vegetation indices results are then subjective evaluation of the exactness of the vegetation cover is performed with the google Earth Pro view of the Study Area.

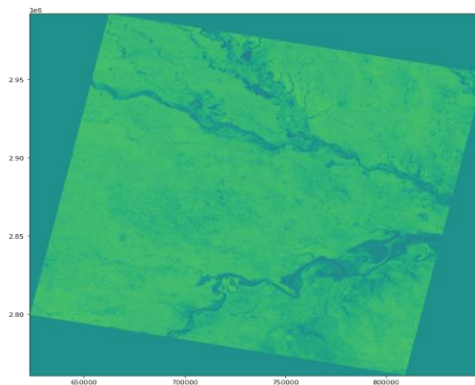
#### **3.1. Detection of Vegetation using NDVI**



**Fig. 5. Results obtained using NDVI**

The rasterio library is used to read the bands of input pre-processed Geotiff images containing the AOI. Using the numpy package in python, NDVI is computed and the resultant image is saved in Geotiff image format. Fig. 6 shows the NDVI image obtained after calculation.

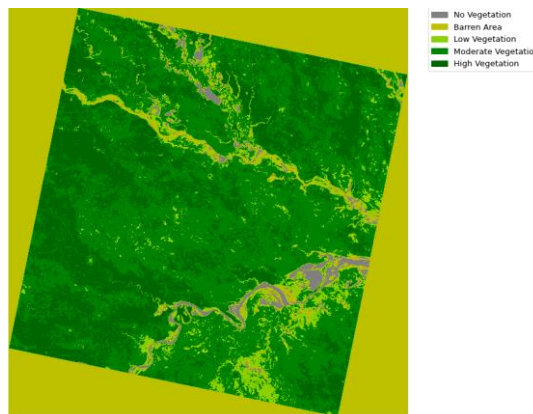
### 3.2. Detection of Vegetation using SAVI



**Fig. 6. Results obtained using SAVI**

The input bands of pre-processed Geotiff image containing the AOI are read using the rasterio library. SAVI is computed using numpy package in python and the resultant image is saved in Geotiff image format. Fig. 7 shows the SAVI image produced after calculation.

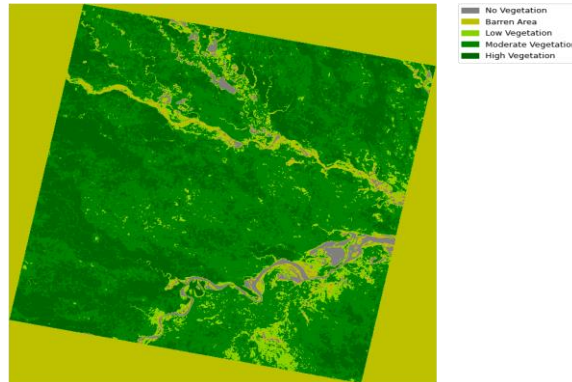
### 3.3. Comparative Analysis of Results



**Fig. 7. Classification result of NDVI**

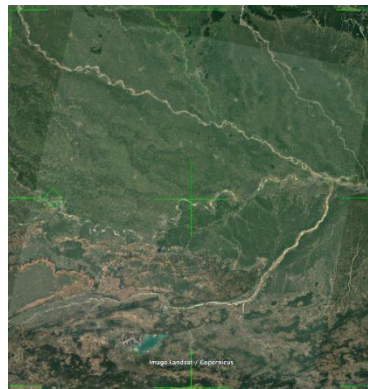
Computation of NDVI involved only spectral measurements from the satellite. SAVI, on the other hand, required the estimation of the L parameter, which is difficult to determine. This is due to the fact that not all soils are alike. Due to many factors such as soil moisture, organic content, mechanical composition, ferrous content, etc., various soils have varied reflectance spectra. NDVI less than 0.25 imply that there is no vegetation present, only soil. Water or urban regions have negative values. The thicker (healthier) the vegetation is, the higher the NDVI values (the same is applicable even for SAVI). However, after a value of 0.7, the NDVI begins to rapidly saturate. As a result, SAVI is more suited to dense vegetation since it saturates at a slower pace. When there is more dense vegetation cover, the L parameter must be modified.





**Fig. 8. Classification result of SAVI**

Clouds, water, and snow contribute negative values, while rocks and barren soil contribute near-zero values. Barren areas correlate to vegetation indices with very low values (0.25 or below) (empty areas of rocks, sand or snow). Small numbers represent low vegetation levels (between 0.25 and 0.4).



**Fig. 9. Google Earth Pro view of the Study Area  
(Credits: Google Earth pro platform)**

Moderate vegetation is represented by values between 0.4 and 0.6, whereas high vegetation is represented by values greater than 0.6. These thresholds are applied on the NDVI and SAVI values and each category of vegetation is represented using different colors as shown in Fig. 8 and Fig. 9. Color grey represents no vegetation, pale yellow (lime) represents bare area, light green represents low vegetation, green represents moderate vegetation, and emerald green represents high vegetation areas (refer Fig. 8 and Fig. 9).

The output of NDVI and SAVI after applying the threshold can be compared by visual inspection with the Google Earth Pro view of the AOI is shown in Fig. 10. The river stream is detected accurately to a greater extent in both NDVI and SAVI. The low vegetation surrounding the river is also identified and when SAVI is used few areas are detected with high vegetation when compared with NDVI.

## 4. CONCLUSION

This paper applied vegetation indices for extracting vegetative features from remotely sensed R and NIR bands of Landsat-8 satellite image. On the pre-processed R and NIR bands of Landsat-8 images acquired on 6<sup>th</sup> September 2013, NDVI and SAVI techniques are applied to determine the characterization of the vegetation cover. The perspective on the study area obtained from Google Earth Engine is utilized to do a subjective evaluation of the exactness of the vegetation cover. This study can be utilized by forestry or urban planning departments to estimate the vegetation cover for LULC. This work can be extended by proposing an Artificial Intelligence (AI) based profound learning model to estimate the progressions in vegetation cover. Further change detection in the AOI can be carried out by applying the vegetation detection methods on multi-temporal satellite images.

## Funding

*This research was not funded by any organization.*

## Acknowledgments

*Authors would like to thank USGS Earth Resources Observation and Science Centre for proving open access to Landsat-8 satellite product also would like to extend thanks to Google Earth pro platform.*

## Conflicts of Interest

*There is no Conflict of Interest.*

## REFERENCES

- Abburu, S., & Golla, S. B. (2015). An engineering evaluation on the glimpse of satellite image pre-processing utility tools. *Article of Engineering Journal*, 19(5), 1–10. <https://doi.org/10.4186/ej.2015.19.2.129>
- Ahmad, A. M., Minallah, N., Ahmed, N., Ahmad, A. M., & Fazal, N. (2020). Remote sensing based vegetation classification using machine learning algorithms. *2019 International Conference on Advances in the Emerging Computing Technologies (AECT)* (pp. 1–6). IEEE. <https://doi.org/10.1109/AECT47998.2020.9194217>
- Asokan, A., Anitha, J., Ciobanu, M., Gabor, A., Naaji, A., & Hemanth, D. J. (2020). Image processing techniques for analysis of satellite images for historical maps classification – an overview. *Applied Sciences*, 10(12), 1–21. <https://doi.org/10.3390/app10124207>
- Bouhennache, R., Bouden, T., & Taleb, A. A. (2014). Change Detection in Urban Land Cover Using Landsat Images Satellites, A Case Study in Algiers Town. *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems* (pp. 622–628). IEEE. <https://doi.org/10.1109/SITIS.2014.57>
- Chen, X., Giri, C., & Vogelmann, J. E. (2016). Land-Cover Change Detection. *Remote Sensing of Land Use and Land Cover* (pp. 152–176). CRC Press. <https://doi.org/10.1201/b11964-14>
- Dalponte, M., Marzini, S., Solano-Correa, Y. T., Tonon, G., Vescovo, L., & Gianelle, D. (2020). Mapping forest wind throws using high spatial resolution multispectral satellite images. *International Journal of Applied Earth Observation Geoinformatics*, 93, 102206. <https://doi.org/10.1016/j.jag.2020.102206>
- Dutta, D., Rahman, A., & Kundu, A. (2015). Growth of Dehradun city: An application of linear spectral unmixing (LSU) technique using multi-temporal landsat satellite data sets. *Remote Sensing Applications: Society and Environment*, 1, 98–111. <https://doi.org/10.1016/j.rsase.2015.07.001>
- Fatihaa, B., Abdelkaderb, A., Latifac, H., & Mohamedd, E. (2013). Spatio temporal analysis of vegetation by vegetation indices from multi-dates satellite images: Application to a semi arid area in ALGERIA. *TerraGreen 13 International Conference 2013 – Advancements in Renewable Energy and Clean Environment, Elsevier, Energy Procedia*, 36, 667–675. <https://doi.org/10.1016/j.egypro.2013.07.077>

- Gandhi, G. M., Parthiban, S., & Christy, N. T. (2015). Ndvi: Vegetation change detection using remote sensing and gis – A case study of Vellore District. *3rd International Conference on Recent Trends in Computing 2015, Elsevier, Procedia Computer Science*, 57, 1199–1210. <https://doi.org/10.1016/j.procs.2015.07.415>
- He, B., Zhang, H., Feng, S., Liu, X., Zhou, Y., & Guan, Y. (2020). Improving land cover change detection and classification with BRDF correction and spatial feature extraction using Landsat Time Series: A case of urbanization in Tianjin, China. *IEEE Journal of Selected Topics in Applied Earth Observation. and Remote Sensing* (vol 13, pp. 4166–4177). IEEE. <https://doi.org/10.1109/JSTARS.2020.3007562>
- Jing, X., Wang, J., Huang, W., Liu, L., & Wang, J. (2009). Study on Forest Vegetation Classification Based on Multitemporal Remote Sensing Images. *Computer and Computing Technologies in Agriculture II, Volume 1. CCTA 2008. IFIP Advances in Information and Communication Technology* (vol 293). Springer. [https://doi.org/10.1007/978-1-4419-0209-2\\_13](https://doi.org/10.1007/978-1-4419-0209-2_13)
- Landsat Missions. (n.d.). Retrieved February 13, 2022 from <https://www.usgs.gov/landsat-missions/landsat-collection-2-level-1-data>
- Langendoen, D., Navarro, G., Willner, W., Keith, D. A., Liu, C., Guo, K., & Meidinger, D. (2020). Perspectives on Terrestrial Biomes: The International Vegetation Classification. *Encyclopedia of the World's Biomes*, 2020, 1–15. <https://doi.org/10.1016/B978-0-12-409548-9.12417-0>
- Li, A., Lei, G., Zhao, W., Nan, X., & Zhang, Z. (2017). Post-earthquake Landslides Mapping from Landsat-8 Data for the 2015 Nepal Earthquake Using a Pixel-Based Change Detection Method. *IEEE Journal of Selected Topics in Applied. Earth Observation and Remote Sensing*, 10, 1758–1768. <https://doi.org/10.1109/JSTARS.2017.2661802>
- Omar, M. S., & Kawamukai, H. (2021). Prediction of NDVI using the Holt-Winters model in high and low vegetation regions: A case study of East Africa. *Scientific African*, 14, e01020. <https://doi.org/10.1007/s40808-018-0431-3>
- Persson, H. J., Ulander, L. M. H., & Soja, M. J. (2018). Modeling and Detection of Deforestation and Forest Growth in Multitemporal TanDEM-X Data. *IEEE Journal of Selected Topic in Applied Earth Observation and Remote sensing*, 11, 3548–3563. <https://doi.org/10.1109/JSTARS.2018.2851030>
- Porikli, F., Bennamoun, M., Khan, S. H., & He, X. (2017). Forest change detection in incomplete satellite images with deep neural networks. *IEEE Transactions on Geoscience and Remote sensing*, 52, 5407–5423. <https://doi.org/10.1109/TGRS.2017.2707528>
- Rhyma, P. P., Norizah, K., Hamdan, O., Faridah-Hanum, I., & Zulfa, A. W. (2019). Integration of normalised different vegetation index and Soil-Adjusted Vegetation Index for mangrove vegetation delineation. *Remote Sensing Applications: Society and Environment*, 17, 1–14. <https://doi.org/10.1016/j.rsase.2019.100280>
- Roy, D. P., Wulder, M. A., Loveland, T. R., Woodcock, C. E., Allen, R. G., Anderson, M. C., Helder, D., Irons, J. R., Johnson, D. M., Kennedy, R., Scambos, T. A., Schaaf, C. B., Schott, J. R., Sheng, Y., Vermote, E. F., Belward, A. S., Bindschadler, R., Cohen, W. B., Gao, F., Hipple, J. D., Hostert, P., Huntington, J., Justice, C. O., Kilic, A., Kovalsky, V., Lee, Z. P., Lymburner, L., Masek, J. G., McCorkel, J., Shuai, Y., Trezza, R., Vogelmann, J., Wynne, R. H., & Zhu, Z. (2014). Landsat-8: Science and product vision for terrestrial global change research. *Remote Sensing of Environment*, 145, 154–172. <https://doi.org/10.1016/j.rse.2014.02.001>
- Ruiz, L. F. C., Guasselli, L. A., Simioni, J. P. D., Belloli, T. F., & Fernandes, P. C. B. (2021). Object-based Classification of Vegetation species in a subtropical wetland using Sentinel-1 and Sentinel-2A images. *Science of Remote Sensing*, 3, 1–10. <https://doi.org/10.1016/j.srs.2021.100017>
- Schmidt, G., Jenkerson, C. B., Masek, J., Vermote, E., & Gao, F. (2013). *Landsat Ecosystem Disturbance Adaptive Processing System (LEDAPS) Algorithm Description* (vi, 9p.). U.S. Geological Survey open-file report, U.S. Geological Survey. <https://doi.org/10.3133/ofr20131057>
- Sowmya, D. R., Deepa, P., & Venugopal, K. (2017). Remote Sensing Satellite Image Processing Techniques for Image Classification: A Comprehensive Survey. *International Journal of Computer Applications*, 161, 24–37. <https://doi.org/10.5120/ijca2017913306>
- Xue, J., & Su, B. (2017). Significant Remote Sensing Vegetation Indices: A Review of Developments and Applications. *Hindawi, Journal of Sensors*, 2017, 1353691. <https://doi.org/10.1155/2017/1353691>
- Yu, M., Xie, Y., & Sha, Z. (2008). Remote sensing imagery in vegetation mapping. *Journal of Plant Ecology*, 1(1), 9–23. <https://doi.org/10.1093/jpe/rtn005>
- Zhu, Z., Fu, Y., Woodcock, C. E., Olofsson, P., Vogelmann, J. E., Holden, C., Wang, M., Dai, S., & Yu, Y. (2016). Including land cover change in analysis of greenness trends using all available Landsat 5, 7, and 8 images: A case study from Guangzhou, China (2000–2014). *Remote Sensing of Environment*, 185, 243–257. <https://doi.org/10.1016/j.rse.2016.03.036>

*Keywords: FEM, Finite Element Method, dynamic analysis, ductile damage, Abaqus*

*Kuba ROŚLANIEC* [0000-0002-4093-1702]\*

# **ANALYSIS OF THE EFFECT OF PROJECTILE IMPACT ANGLE ON THE PUNCTURE OF A STEEL PLATE USING THE FINITE ELEMENT METHOD IN ABAQUS SOFTWARE**

## **Abstract**

*This paper deals with the punctureability of a steel plate by a projectile at different angles of attack. The effect of the projectile angle on the force required to penetrate a plate made of A36 steel is presented using Finite Element Method calculation software. Using Abaqus software, a dynamic model of a projectile striking a plate was modelled and the force required to penetrate a 5 mm thick steel plate was presented. The introduction gives an overview of the genesis of the topic and a brief historical background. The chapter on numerical analysis presents the numerical model used and how the simulation was modelled. In the conclusions, a summary of the results was formulated and conclusions were drawn regarding the observations and insights of the analysis. The force required to penetrate the plate was observed to increase with increasing projectile angle of attack and it was found that, as the angle of the plate increased, the force required to penetrate increased.*

## **1. INTRODUCTION**

The problem of steel plate penetrability became widespread when the first tank appeared on the battlefield, the British Mark I. The concept of armoured weapons was then born, and with it the idea of destroying them (Fletcher, 2004).

The ability to penetrate armour depends on the properties of the armour, the properties of the projectile (material from which it was made, design, type of detonator) and the velocity, angle of attack of the projectile and the shape of its nose. In the article "Ballistic resistance of 2024 aluminium plates against hemispherical, spherical and blunt nose projectiles", it was observed that the lowest permeability is characteristic of projectiles with a spherical nose in comparison with projectiles with a hemispherical and flat nose. At low velocities, flat nose projectiles had the highest global energy (Senthil et al, 2018).

---

\* Lublin University of Technology, Poland

In order to improve the vehicle's resistance to bullet penetration, the thickness of the armour was increased, resulting in an increase in weight, and a deterioration in the vehicle's mobility and manoeuvrability. For this reason, research began into more puncture-resistant materials, and due to the impossibility of carrying out non-destructive tests, experiments were carried out by shooting at armour plates with various types of weapons. Figure 1 shows a ballistic plate which was tested on a firing range (Torecki, 1982). In the course of experimental research, extended by the achievements of modern technology and computer simulation medotypes, it was observed that, in the case of thin plates (in relation to the diameter of the projectile), when the local deformation was twice as large as the diameter of the plate, almost no global deformation was registered, in the case of thicker plates this phenomenon is not observed. With increasing plate thickness and impact velocity, an increase in projectile deformation is observed. It was also found that the perforation time of the plate is relatively constant for different plate thicknesses at impact velocities close to the limit velocity required to perforate the material. In the case of not piercing the metal plate, exceeding the ballistic limit, projectile cracking was observed causing minor damage to the plate (Børvik et al, 2003).

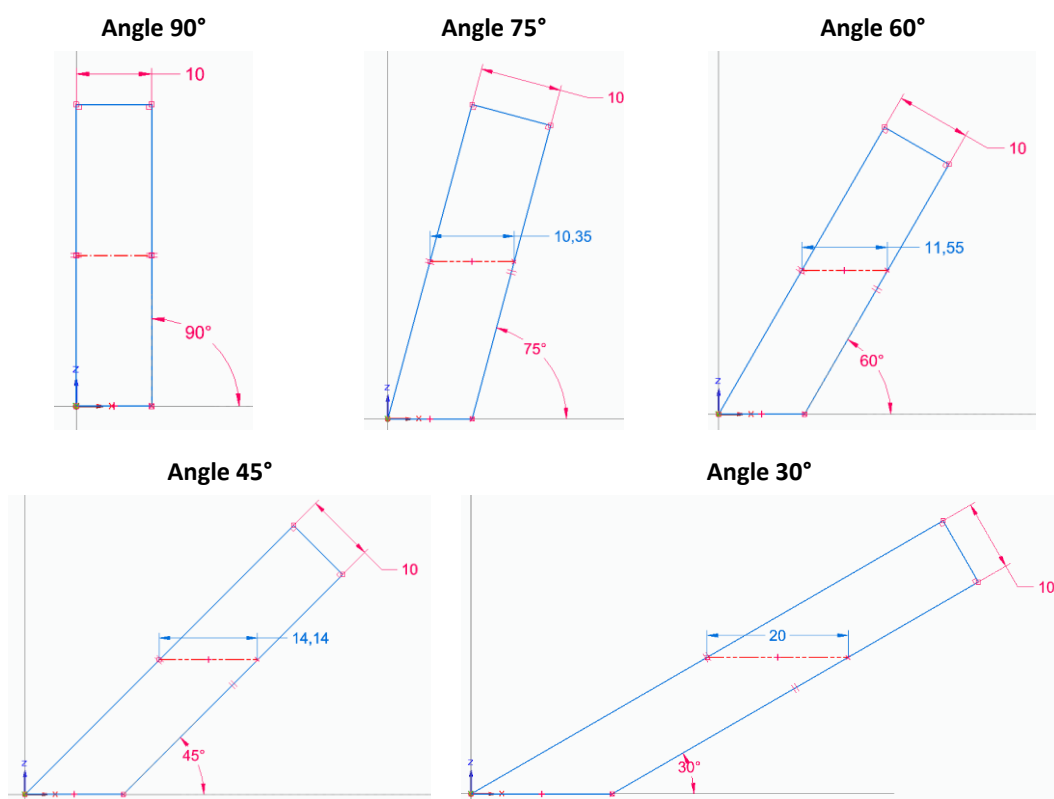
Ballistic tests showed that the highest resistance to penetration by conventional armour-piercing bullets characterised the plates made steels with a significant content of alloying elements, such as Mo-molybdenum (eliminates brittleness), Cr-chromium (increases hardness), Ni-nickel (increases yield strength, which is an important prerequisite for strength, in conventional bullets). Due to the complexity of the production process and the high price of alloying elements, armour plates made of alloy steels were significantly more expensive than their non-alloy counterparts. In the 1930s, the Soviet Union commissioned the Kharkov Steam Engine Factory to develop a universal tank. Mikhail Koshkin's team observed during the testing of the prototype BT-SV tank, as illustrated in the Figure 3, the beneficial effect of setting the armour plates at an angle (Michulec & Zientarzewski, 2006; Flis & Sperski, 2012).



**Fig. 1.** View of a section of an 80 mm thick plate under test firing (Aufmann, 2020)

As the angle of inclination of the armour increases, the effective thickness of the armour (i.e., the thickness of the armour through which the bullet must penetrate to pierce the armour) is greater. The dependence of the effective armour thickness on the angle of inclination is shown in Figure 2. Furthermore, compared to a projectile striking a plate positioned perpendicular to the angle of attack of the projectile, where the entire kinetic energy of the projectile is transferred to the armoured plate, there is a partial dissipation of this energy, and there is an increased chance of the projectile rebounding, sliding off the vehicle plate, the so-called ricochet (Torecki, 1982; Michulec & Zientarzewski, 2006). When the projectile hits the plate, the kinetic energy of the system decreases, it is converted into work needed to overcome the force of resistance and work of deformation. It has also been observed that the most favourable situation is that the structure of the armour plate is not damaged by the projectile, which results in the ballistic protection of the plate not being affected. It should be borne in mind that in certain situations increasing the angle of inclination of the plate is not a fully effective method of protection, due to the possibility of the aforementioned ricochets causing damage to other people or other vehicles (Nieoczym & Drozd, 2021).

In this paper, a numerical analysis is carried out for a homogeneous steel plate that has constant physical and chemical properties throughout its cross-section. These plates are the least advanced type of steel plates used in military facilities and vehicles, however, due to their relatively low price, and easy manufacturing process, they were widely used in the 20th century (Michulec & Zientarzewski, 2006; Flis & Sperski, 2012).



**Fig. 2. Effect of armour plate inclination angle on effective armour thickness**





**Fig. 3. View of a section of an 80 mm thick plate under test firing (Czołg lekki BT-SV-2 „Turtle”, n.d.)**

## **2. NUMERICAL ANALYSIS**

In the analytical method, Newton's laws are used to describe the motion of the material particles pierced by the projectile, this motion has the character of a plasticised movement of the material particles in three-dimensional space. This leads to the formulation of complex systems of molecular differential equations. Attempts to estimate by analytical means the simple case of armour piercing by a projectile have a lower accuracy than numerical analyses, as the level of complexity of the model increases, the inaccuracy of the analytical model increases. However, analytical methods can be used for preliminary assessment and analysis of the correctness of finite element calculations. In addition, computer simulation provides information about the deformations and stresses produced during the material puncture test (Budzyński, 2006; Wiśniewski & Żurowski, 2002)

The essence of the Finite Element Method (FEM) is the approximation of a continuous medium with an infinite number of degrees of freedom by using a discrete model with a finite number of degrees of freedom. The foundations of the Finite Element Method were published in a paper by Kirschhof in 1868. He proposed to replace the three-dimensional system a continuous spatial truss composed of cuboid elements. The revival of the above idea took place after the Second World War, thanks to Polish scientists – Professor Olgierd Zienkiewicz and Professor Janusz Przemieniecki. They tackled the problem of the excessive number of unknowns in equilibrium systems. Prof. Zienkiewicz is regarded as the creator and forerunner of the Finite Element Method and its applications in mechanics. The use of this method involves solving equations with thousands of unknowns (Bathe, 1996; Flis & Sperski, 2011).

The ductile damage initiation criterion was adopted to estimate the effect of projectile angle on plate penetrability. It is used to predict the onset and expansion of damage caused by nucleation, growth and coalescence of voids in ductile metals. The ductile damage initiation criterion was adopted to estimate the effect of projectile angle on plate penetrability.

It is used to predict the onset and expansion of damage caused by nucleation, growth and coalescence of voids in ductile metals. This model assumes that, the onset of plastic deformation is equivalent (is equivalent) to a function of stress triaxiality and strain rate (Simulia, 2017; Wagner, 2021).

$$\bar{\epsilon}_D^{pl} = d_0 e^{-c\mu} + d_1 e^{c\mu} \quad (1)$$

where:  $\bar{\epsilon}_D^{pl}$  – stress-strain rate triaxial function,

$\mu = \frac{-p}{q}$  – stress triaxiality function,

$p$  – pressure stress,

$q$  – mises equivalent stresses,

$c, d_0, d_1$  – aterial characteristics obtained from research and testing.

All equations included in the paper are taken from (Simulia, 2017).

The application of this computational model requires advanced research and testing, due to its significant cost, labour intensity and difficulty of implementation, the article refers to the results of other researchers. The material properties of the steel were obtained from tensile coupon tests as described in the experimental study: "More Details about Validation of Calibrated Parameters of Ductile Damage Model for A36 Steel Plates in Abaqus Software" (Tiwari, Iqbal & Gupta, 2018). The penetration simulations were carried out for a plate made of A36 steel. It is a high strength structural carbon steel used for hot rolled plates, shaped bars and universal profiles. It is characterised by good weldability to ASTM standards, an important feature in the manufacture of armoured vehicles (Steel Construction Manual, 1986).

**Tab. 1. Percentage chemical composition of A36 steel in ASTM standard (Steel Construction Manual, 1986)**

<b>C</b>	<b>Si</b>	<b>Mn*</b>	<b>S</b>	<b>P</b>	<b>Cu</b>
≤ 0.25	≤ 0.40	–	≤ 0.05	≤ 0.04	≥ 0.20

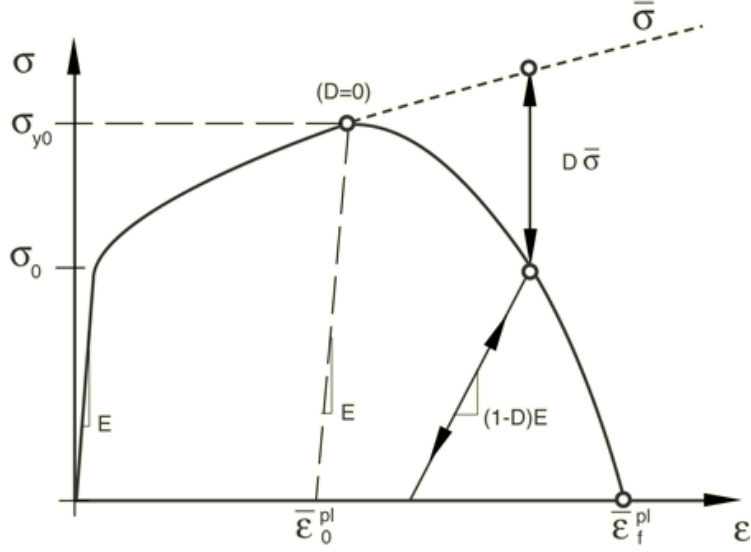
\*The manganese content is not regulated.

**Tab. 2. Physical and mechanical properties in analysis (Steel Construction Manual, 1986)**

<b>Tensile Strength, Ultimate</b>	550 MPa
<b>Plastic Strain</b>	0.2
<b>Tensile Strength, Yield</b>	250 MPa
<b>Modulus of Elasticity</b>	200 GPa
<b>Poissons Ratio</b>	0.26
<b>Shear Modulus</b>	79.3 GPa
<b>Density</b>	7.85 g/cm <sup>3</sup>

For damage, the Damage Evolution sub-option has been added to the material model. Fracture strain is an equivalent strain at damage initiation. Figure 4 shows the relationship between stress and strain that a material undergoes in during failure – a continuous line, while the dashed curve is the relationship in the absence of damage. The displacement value at failure corresponds to the product of the characteristic length of the elements and the strain at failure (Wagner, 2021; Askariani & Garivani, 2020).





**Fig. 4. Stress-strain curve plot with advancing damage degradation (Simulia, 2017)**

The stress triaxiality is defined as ratio pressure stress  $-p$  and the mises equivalent stress  $-q$  (Simulia, 2017; Wagner, 2021).

$$\eta = \frac{-p}{q} \quad (2)$$

The strain rate is an equivalent plastic strain rate. The strain rate is the deformation, the deformation of a material over time, due to a load (Simulia, 2017; Wagner, 2021).

$$\epsilon(t) = \frac{L(t) - L_0}{L_0} \quad (3)$$

$$\dot{\epsilon} = \frac{d\epsilon}{dt} = \frac{d}{dt} \left( \frac{L(t) - L_0}{L_0} \right) = \frac{1}{L_0} \times \frac{dL(t)}{dt} = \frac{v(t)}{L_0} \quad (4)$$

Displacement damage evolution is defined as a function of the total, for cohesive elements or bulk elastic-plastic materials, displacement at the time of damage initiation. The value of the failure properties adopted are shown in Table 3 and Figure 4 (Wagner, 2021).

**Tab. 3. The Required values of Ductile Damage model for A36 steel plates (N, mm), (Askariani & Garivani, 2020).**

No.	Fracture Strain	Stress Triaxiality	Strain Rate	No.	Fracture Strain	Stress Triaxiality	Strain Rate
1	3.4285	0.01	1	20	2.3207	0.2	1
2	3.3587	0.02	1	21	2.2736	0.21	1
3	3.2903	0.03	1	22	2.2275	0.22	1
4	3.2233	0.04	1	23	2.1823	0.23	1
5	3.1577	0.05	1	24	2.1381	0.24	1
6	3.0934	0.06	1	25	2.0948	0.25	1
7	3.0305	0.07	1	26	1.8913	0.3	1
8	2.9688	0.08	1	27	1.5430	0.4	1
9	2.9084	0.09	1	28	1.2606	0.5	1
10	2.8493	0.1	1	29	1.0322	0.6	1
11	2.7913	0.11	1	30	0.8480	0.7	1
12	2.7346	0.12	1	31	0.7001	0.8	1
13	2.6790	0.13	1	32	0.5821	0.9	1
14	2.6246	0.14	1	33	0.4891	1	1
15	2.5712	0.15	1	34	0.4169	1.1	1
16	2.5190	0.16	1	35	0.3626	1.2	1
17	2.4679	0.17	1	36	0.3239	1.3	1
18	2.4178	0.18	1	37	0.2990	1.4	1
19	2.3687	0.19	1	38	0.2868	1.5	1

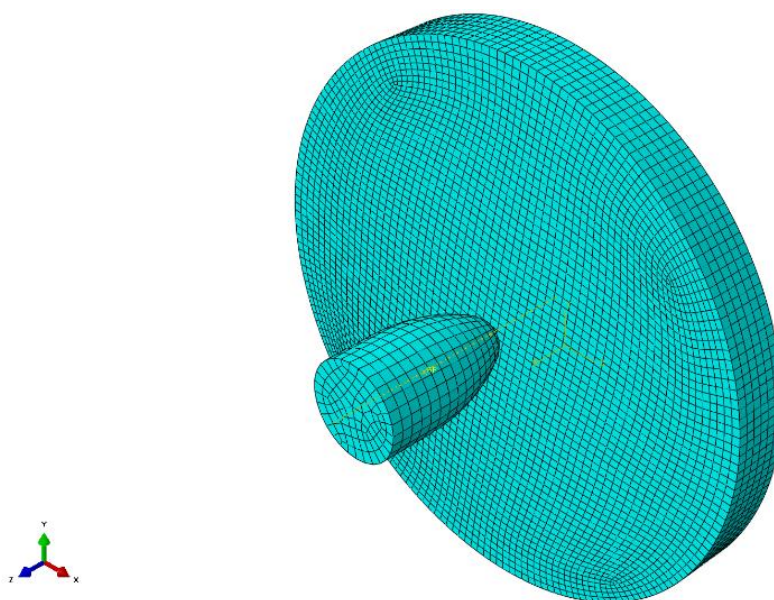
Displacement at Failure	
1	0.3

**Fig. 5. Damage Evolution suboption settings in the Ductile Damage material model for A36 steel plates (N, mm) (Tyutin, Botvinaa & Petersen, 2020)**

In this study, the numerical simulation was carried out in Abaqus software, dedicated to carrying out advanced simulations using the Finite Element Method. The computational model consists of two elements, the object of analysis being a steel plate made of A36 steel, shot through by a projectile at various angles of attack, with a constant velocity of  $V = 10000$  mm/s, along the Z axis relative to the reference system, in order to eliminate unfavourable projectile displacement, projectile movement in the X and Y planes is blocked. Steel plate restrained by locking the possibility of movement in the X, Y, Z plane on the lateral edge of the disc. The model of the shot-peened slab (a disc of 50 mm diameter and 5 mm thickness) was designed as a deformable solid object, with a Mesh type C3D8R (An 8-node linear brick, reduced integration, hourglass control), composed of 12100 finite elements. The slab was shot through at the following angles of attack:

- 90° (effective thickness -5 mm),
  - 75° (effective thickness -5.18 mm, increased thickness over 90° by 3.5%),
  - 60° (effective thickness -5.77 mm, increased thickness over 90° board by 15.5%),
  - 45° (effective thickness -7.07 mm, increased thickness over 90° board by 41.5%),
  - 30° (effective thickness -10 mm, increased thickness over 90° board by 100%),
- (Różyło, 2020, Ep. 1. and Ep. 2.).

In order to accelerate the performed calculations, the bullet model was designed as a non-deformable shell object, type Discrete rigid, base feature Shell, with a mesh type R3D4 (a 4-node 3-D bilinear rigid quadrilateral) composed of 700 finite elements. The bullet geometry is inspired by the  $9 \times 19$  mm Parabellum pistol cartridge (Nieoczyn & Drozd, 2021).



**Fig. 6. Composition and Mesh of the numerical model tested for 0° angle of attack**

The simulations were modelled as a dynamic analysis, taking into account geometric nonlinearities resulting from significant deformations. Due to the time-consuming nature of dynamic analyses, the duration of the analysis was limited, Time Period: 0.01, and in the model assembly the projectile was oriented at a short distance from the plate at the start of the analysis. Additionally, mass scaling was applied (Scale by factor = 600), this procedure significantly reduces the duration of the numerical analysis. The value of mass scaling, according to the principle of process acceleration, influences the course of the analysis and the results obtained. However, in the conducted analysis, the effect of the projectile angle of attack on the puncture is studied, which is not affected by the mass scaling phenomenon, as the analyses compared with each other were performed with the same scaling factor (Różyło, 2020a, 2020b).

### 3. RESULTS OF ANALYSIS

The highest stresses in the plate for all impact variants are observed on the projectile exit side in the axis to the direct point of contact of the projectile nose with the plate and in its vicinity. The occurrence of stress zones, i.e. places where the stress values are close to each other, was observed. During penetration of the plate by the projectile, propagation of stresses is observed, progressing from the point of impact, the zone of the highest stresses constantly increases until the hole, created as a result of the hit, reaches a value corresponding to or greater than the largest diameter of the projectile, then gradually the stresses reduce, producing the characteristic zones visible in the visualisations below. The maximum stress value was:  $S\text{-Misses}_{\max} = 550 \text{ MPa}$ .

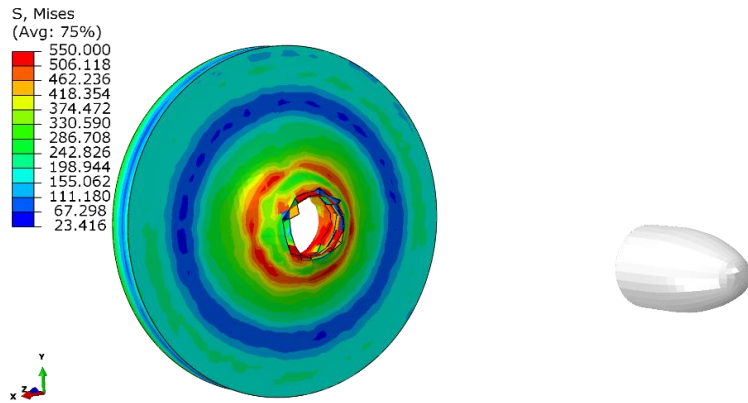
In the case of the  $90^\circ$  angle, the distribution of the values of the individual stresses assumes a characteristic shape close to circles, whose centre is the point of first contact of the projectile with the plate. The penetration surface, a fragment of the plate completely ruptured by the projectile, also assumes the shape of a circle. These stresses are characterised by a significant level of symmetry, relative to the individual planes. In Figure 7 it is possible to notice two zones of sudden decrease of stresses, the first one occurring near the centre of the plate, created between the zones of plasticised material, and the second one created near the place of creation of global deformations, this phenomenon can be noticed in all variants of simulations. In order to verify the statement concerning the second zone, it is necessary to conduct additional tests, for a larger diameter of the plate, due to the proximity of the outer edge of the plate to this zone.

For angles of  $75^\circ$ ,  $60^\circ$  and  $45^\circ$ , shows in a Figure 8–11, a deformation of the circles of the stress zones was observed in relation to the angle of attack of  $90^\circ$ . As the angle of attack increases, the distribution of these zones becomes less ordered, the stresses are less regularly distributed and take on increasingly distorted forms. Moreover, as the slope of the plate decreases, the zone of maximum stresses is closer to the point of impact of the projectile. For an angle of  $75^\circ$ , the stress zones take the shape of ellipses flattened along the horizontal plane. In the case of  $30^\circ$  projectile impact angle, it is not possible to isolate the stress zones due to the too small diameter of the tested plate; in order to observe their distribution, additional simulation must be carried out. The observed stress zones have irregular, strongly deformed shapes. The characteristic circles and ellipses found in the other variants were also not observed, but this finding cannot be unequivocally confirmed.

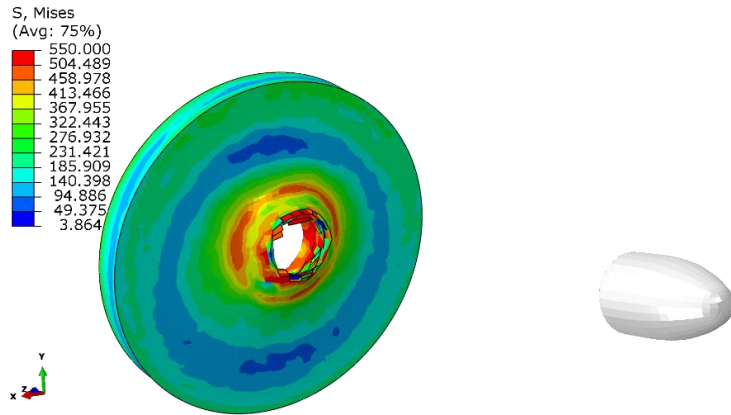
During the penetration of the plate by the projectile, observations were made regarding the maximum displacements in the plate, these were as follows:

- for an angle of  $90^\circ$ ,  $U_{\max} = 11.580 \text{ mm}$ ,
- for  $75^\circ$  angle,  $U_{\max} = 11.738 \text{ mm}$ ,
- for  $60^\circ$ ,  $U_{\max} = 10.211 \text{ mm}$ ,
- for  $45^\circ$ ,  $U_{\max} = 10.756 \text{ mm}$ ,
- for a  $30^\circ$  angle,  $U_{\max} = 10.328 \text{ mm}$ .

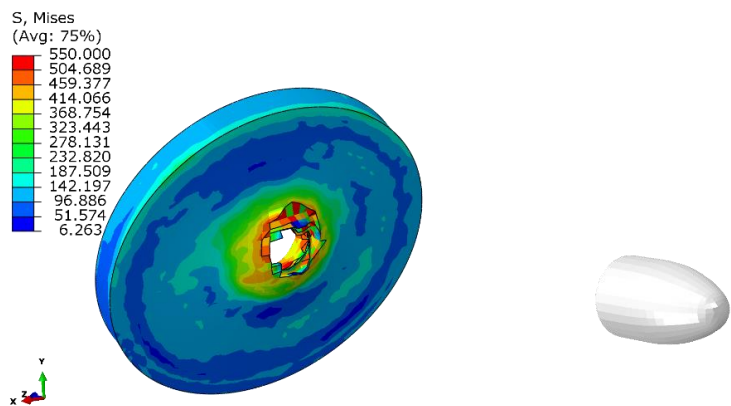
It was found that there was no dependence of the maximum deformation values on the projectile angle of attack. The highest deformation was obtained for an angle of  $75^\circ$ , while the lowest for an angle of  $60^\circ$ . In order to know the exact value of the displacement at which the maximum force for the puncture occurs, an additional analysis with a smaller calculation step must be carried out.



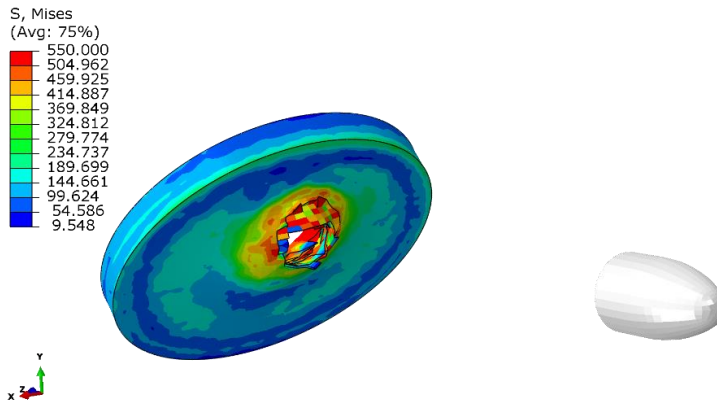
**Fig. 7. Visualisation of the projectile impact on the plate, angle of attack 90°**



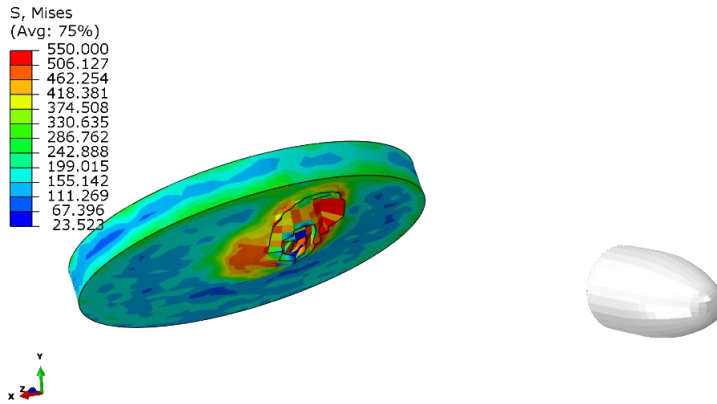
**Fig. 8. Visualisation of the projectile impact on the plate, angle of attack 75°**



**Fig. 9. Visualisation of the projectile impact on the plate, angle of attack 60°**



**Fig. 10. Visualisation of the projectile impact on the plate, angle of attack 45°**



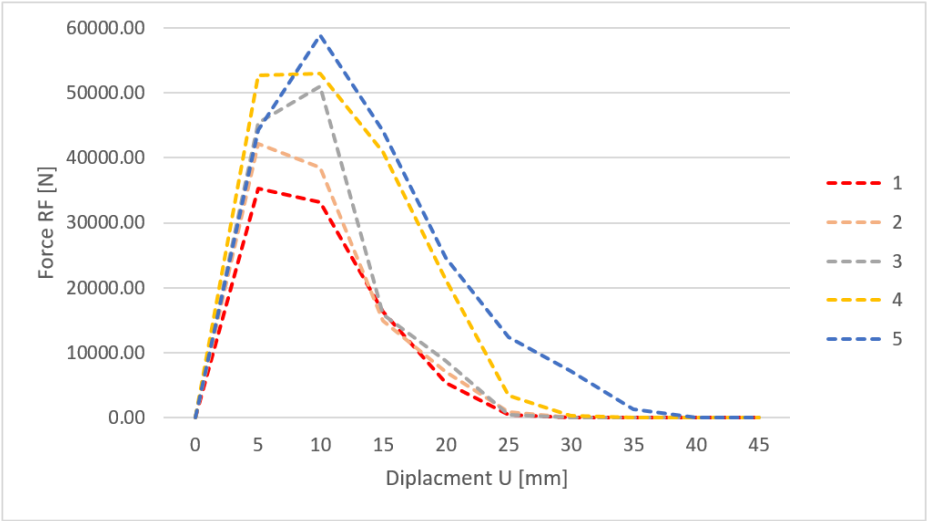
**Fig. 11. Visualisation of the projectile impact on the plate, angle of attack 30°**

Table 4 shows the values of forces obtained in the simulation against displacement. The highest value of the force needed for penetration was obtained for the angle of 30°, which is 58801.9 [N], while the lowest for the angle of 90°, which is 35211.2 [N]. Analysing the results, it is possible to notice an increase in the force required to penetrate the projectile with an increase in the angle of attack of the projectile. This is due to an increase in the effective thickness of the armour, resulting in a longer distance for the projectile to travel in the plate. For the first and second cases the maximum force value is observed for a displacement of 5 millimetres, for the other cases for a displacement value of 10 millimetres. The dependence of the maximum force on the displacement is shown in Figure 12. The change in the displacement value at which the maximum stress occurs is due to the fact that, as the angle of attack increases, the projectile travels a longer distance until the upper and lower surfaces of the nose contact the plate. Figure 13 shows the value of the maximum force required for penetration for all variants. The area of the surface in contact with the plate has a significant influence on the force needed for penetration and the value of the displacement. Analysing the graph, an almost linear tendency of the increase of the force required for

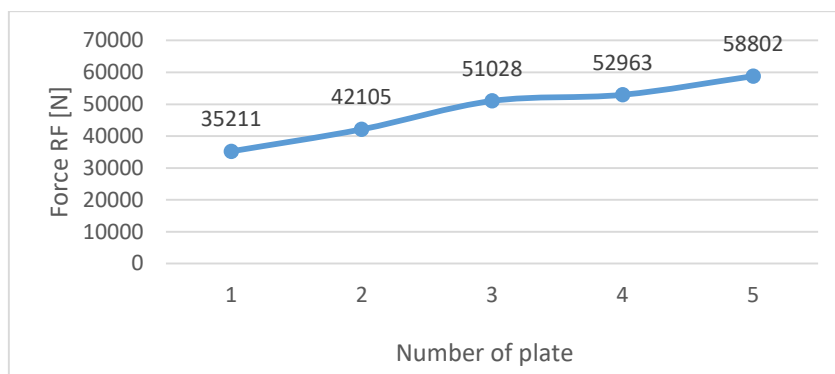
puncture was observed with the change of the slab inclination angle. Moreover, no proportionality of the increase in the force necessary for penetration with respect to the change in the effective thickness of the plate was observed. The lack of proportionality is due to a change in the area of the projectile in contact with the plate during penetration (Rodriguez-Millan et al., 2018).

**Tab. 4. Table of results obtained in the numerical analysis**

Number of simulation	1.		2.		3.		4.		5.	
Rake Angle	90°		75°		60°		45°		30°	
No. \	U [mm]	RF [N]	U [mm]	RF [N]	U [mm]	RF [N]	U [mm]	RF [N]	U [mm]	RF [N]
1.	0	0.00	0	0	0	0	0	0	0	0
2.	5	35211.20	5	42105.4	5	45387.3	5	52729	5	44291
3.	10	33237.60	10	38472.1	10	51027.9	10	52963	10	58801.9
4.	15	16296.40	15	14929.1	15	15955.4	15	40900.9	15	44187.8
5.	20	5364.43	20	7135.66	20	8789.04	20	21304.9	20	24555.6
6.	25	495.19	25	939.91	25	481.784	25	3427.11	25	12404.3
7.	30	0.00	30	0	30	0	30	268.788	30	7271.49
8.	35	0.00	35	0	35	0	35	0	35	1364.85
9.	40	0.00	40	0	40	0	40	0	40	0



**Fig. 12. Diagram of the dependence of the force required for penetration on the angle of attack of the bullet**



**Fig. 8. Maximum punching force value for individual plates**

#### 4. CONCLUSIONS

On the basis of the numerical analysis carried out in the Abaqus software, the following conclusions were drawn concerning the impact of the projectile on the steel plate.

1. Increasing the angle of attack of a projectile striking a plate, increases the force required to penetrate the plate. For a plate set at  $90^\circ$ , the force required to penetrate the plate increases by a value:

- 19.0% for a plate positioned at an angle of  $75^\circ$ ,
- 44.9% for a plate positioned at an angle of  $60^\circ$ ,
- 50.4% for a plate positioned at an angle of  $45^\circ$ ,
- 67% for a plate positioned at an angle of  $30^\circ$ .

2. The value of the force required for penetration increases approximately linearly with increasing angle of attack, and the largest increase in the energy required for penetration, relative to the previous position of the plate, is observed at an angle of  $60^\circ$  and is 25.9% relative to a plate positioned at  $75^\circ$ .

3. The elastic-plastic deformations were observed during the simulations, the highest value of plastic deformations is observed at the projectile impact site. The highest value of elastic deformation was observed on the plate wall opposite to the impact, in the vicinity of the projectile impact site.

4. For angles of  $75^\circ$ ,  $60^\circ$ ,  $45^\circ$  the increase in force required for penetration is greater than would be indicated by the percentage increase in armour thickness. The greatest differences were observed for the  $60^\circ$  angle, where the effective armour thickness increased by 15.5% and the force required for penetration increased by 44.9%, relative to the  $90^\circ$  plate.

In conclusion, the angle of attack of the plate is an important aspect influencing the strength of the plate when hit by a projectile. By increasing the angle of attack, safety can be improved without increasing the weight or changing the armour material. It should be borne in mind that the use of armour made up of inclined plates reduces the usable area of the vehicle. The values and percentage differences obtained are adequate for projectiles with a hemispherical tip. In order to assess the incremental force required to penetrate, as the angle of attack increases, for projectiles with other vertices (e.g. cylindrical, conical), additional numerical analysis must be carried out. However, it can be concluded that for all types of projectile tips, increasing the angle of attack increases the force required to penetrate a plate made of homogeneous isotropic material.



During the analysis of the results, the need for complementary analyses was identified. Such as:

- conducting analyses for a larger disc diameter, in order to know more precisely the stress distribution,
- carrying out the analysis for a smaller calculation step in order to know the exact displacement values for which the maximum stresses are obtained.

Due to the complexity of the topic related to the penetration of the plate by the projectile, additional analyses should be considered to complement the state of the art in ballistics, e.g. the effect of projectile velocity on the ability to penetrate a steel plate.

## REFERENCES

- Askariani, S. S., & Garivani, S. (2020). *More Details about Validation of Calibrated Parameters of Ductile Damage Model for A36 Steel Plates in Abaqus Software University of Bojnord*. University of Bojnord.
- Aufmann, F. (2020 March, 17). Stahl – und Panzer-Platten – Stalowe i pancerne plyty. <http://hauba.pl/stahl-und-panzer-platten-stalowe-i-pancerne-plyty>
- Bathe, K. J. (1996). *Finite Element Procedures*. Prentice Hall.
- Børvik, T., Hopperstada, O. S., Langsetha, M., & ArneMalloa, K. (2003). Effect of target thickness in blunt projectile penetration of Weldox 460 E steel plates. *International Journal of Impact Engineering*, 28(4), 413–464. [https://doi.org/10.1016/S0734-743X\(02\)00072-6](https://doi.org/10.1016/S0734-743X(02)00072-6)
- Budzyński, A. (2006). Krótki wstęp do zastosowania Metody Elementów Skończonych (MES) do numerycznych obliczeń inżynierskich. *Biuletyn „GM View”*, 5/2006.
- Czołg lekki BT-SV-2 „Turtle”. (n.d.). Retrieved March 16, 2020 from <https://pl.ww2facts.net/28359-light-tank-bt-sv-2-turtle.html>
- Fletcher, D. (2004). *British Mark I Tank 1916*. Osprey Publishing.
- Flis, L., & Sperski, M. (2011). Badania odporności balistycznej pancerzy ze stali 10GHMBA na ostrzał pociskami 12.7 mm. *Zeszyty naukowe Akademii Marynarki Wojennej*, 3(186), 27–42.
- Flis, L., & Sperski, M. (2012). Ocena wpływu kształtu wierzchołka pocisku na proces przebijania pancerzy stalowych. *Zeszyty naukowe Akademii Marynarki Wojennej*, 2(189), 29–44.
- Michulec, R., & Zientarzewski, M. (2006). *T-34 – Mityczna broń* (pp. 12–40). Wydawnictwo Armagedon.
- Nieoczym, A., & Drozd, K. (2021). Fractographic Assessment and FEM Energy Analysis of the Penetrability of a 6061-T Aluminum Ballistic Panel by a Fragment Simulating Projectile. *Advances in Science and Technology Research Journal*, 15(1), 50–57. <https://doi.org/10.12913/22998624/129951>
- Rodriguez-Millan, M., Garcia-Gonzalez, D., Rusinek, A., & Arias, A. (2018). Perforation mechanics of 2024 aluminium protective plates subjected to impact by different nose shapes of projectiles. *Thin-Walled Structures*, 123, 1–10.
- Różyło, P. (2020a). *Metoda elementów skończonych Praktyczne przykłady zagadnień statycznych i dynamicznych w programie Abaqus: Część 1*. Wydawnictwo Politechniki Lubelskiej.
- Różyło, P. (2020b). *Metoda elementów skończonych Praktyczne przykłady zagadnień statycznych i dynamicznych w programie Abaqus: Część 2*. Wydawnictwo Politechniki Lubelskiej.
- Senthil, K., Iqbal, M. A., Arindam, B., Mittal, R., & Gupta, N. K. (2018). Ballistic resistance of 2024 aluminium plates against hemispherical sphere and blunt nose projectiles. *Thin Walled Structures*, 126, 94–105.
- Simulia, D. (2017). *Abaqus 2017 documentation*. Dassault Systemes Waltham.
- Steel Construction Manual* (8th Edition, second revised edition). (1986). American Institute of Steel Construction.
- Tiwari, G., Iqbal, M. A., & Gupta, P. K. (2018). Energy absorption characteristics of thin aluminium plate against hemispherical nosed projectile impact. *Thin-Walled Structures*, 126, 246–257.
- Torecki, S., (1982). Przebijalność pocisku. *W 1000 słów o broni i balistyce* (pp. 177–179). Wydawnictwo Ministerstwa Obrony Narodowej.
- Tyutin, M. R., Botvinaa, L. R., & Petersen, T. B. (2020). Tensile damage evolution of structural steels with different structure. *Procedia Structural Integrity*, 28, 2148–2156. <https://doi.org/10.1016/j.prostr.2020.11.042>
- Wagner, R. H. N. (2021). *ABAQUS Tutorial: Damage for Ductile Metals – Material Model Explained – Ductile Damage*. Project: ABAQUS – Material Model Explained – Video Series.
- Wiśniewski, A., & Żurowski, W. (2002). *Amunicja i pancerze*. Politechnika Radomska.

*Keywords: coronary heart disease, machine learning, ensembles, outlier detection, framingham*

*Lubna RIYAZ* [0000-0002-4502-7720]\*, *Muheet Ahmed BUTT* [0000-0002-8059-0180]\*, *Majid ZAMAN*\*\*

## IMPROVING CORONARY HEART DISEASE PREDICTION BY OUTLIER ELIMINATION

### Abstract

Nowadays, heart disease is the major cause of deaths globally. According to a survey conducted by the World Health Organization, almost 18 million people die of heart diseases (or cardiovascular diseases) every day. So, there should be a system for early detection and prevention of heart disease. Detection of heart disease mostly depends on the huge pathological and clinical data that is quite complex. So, researchers and other medical professionals are showing keen interest in accurate prediction of heart disease. Heart disease is a general term for a large number of medical conditions related to heart and one of them is the coronary heart disease (CHD). Coronary heart disease is caused by the amassing of plaque on the artery walls. In this paper, various machine learning base and ensemble classifiers have been applied on heart disease dataset for efficient prediction of coronary heart disease. Various machine learning classifiers that have been employed include k-nearest neighbor, multilayer perceptron, multinomial naïve bayes, logistic regression, decision tree, random forest and support vector machine classifiers. Ensemble classifiers that have been used include majority voting, weighted average, bagging and boosting classifiers. The dataset used in this study is obtained from the Framingham Heart Study which is a long-term, ongoing cardiovascular study of people from the Framingham city in Massachusetts, USA. To evaluate the performance of the classifiers, various evaluation metrics including accuracy, precision, recall and f1 score have been used. According to our results, the best accuracy was achieved by logistic regression, random forest, majority voting, weighted average and bagging classifiers but the highest accuracy among these was achieved using weighted average ensemble classifier.

### 1. INTRODUCTION

Heart is a vital organ in the human body performing the crucial function of pumping blood to different parts of the body. A slight change in the normal functioning of the heart can lead to imbalance in the functioning of the whole body. Heart diseases also known as the cardiovascular diseases are the conditions that affect the structure or function of our

---

\* PG Department of Computer Sciences, University of Kashmir, Srinagar, India, lubna.riyaz122@gmail.com, ermuheet@gmail.com

\*\* Directorate of IT & SS, University of Kashmir, Srinagar, India, zamanmajid@gmail.com

heart (Cardiovascular (Heart) Diseases: Types and Treatments, n.d.). Heart diseases can be caused by various factors such as lifestyle changes including bad eating habits, sleep deprivation, less physical activity, smoking habits, etc. According to a report by World Health Organization, non-communicable diseases including heart diseases, strokes and other diseases are collectively responsible for death of almost 41 million people each year (Less than \$1: How WHO Thinks That Can Save 7 Million Lives, n.d.). The various symptoms of heart diseases include chest pain, chest tightness, chest pressure and chest discomfort (Heart Disease – Symptoms and Causes – Mayo Clinic, n.d.). There are different kinds of heart diseases that include blood vessel disease, such as coronary artery disease, heart rhythm problems (arrhythmias), heart defects that some people are born with (congenital heart defects), heart valve disease, disease of the heart muscle, heart infection, etc. (Heart Disease – Symptoms and Causes – Mayo Clinic, n.d.). So, detection of heart disease at an early stage can help in saving numerous human lives. Detection of heart disease depends on blend of pathological and clinical data that is complex in nature. Today, medical industries produce huge quantities of data related to patients' health. This data needs to be processed to obtain useful information from it.

Heart disease is a general term for a large number of medical conditions related to heart and one of them is the coronary heart disease. Coronary heart disease is caused by the accumulation of fatty deposits on artery walls around heart and can result in severe illness and even death of the patient. According to a report by NHS (National Health Service), UK, coronary heart disease is a major cause of death in UK and worldwide (Coronary Heart Disease – NHS, n.d.). Various symptoms of coronary heart disease include shortness of breath, body pain, chest pain, feeling nausea, etc. With the passage of time, artery walls get furred up because of fatty substance known as atheroma (Coronary Heart Disease - NHS, n.d.). It can be caused by various lifestyle changes including drinking excessive alcohol, smoking etc. The build-up of fatty deposits on artery walls can block the blood flow (totally or partially) to the heart (Coronary Heart Disease | NHLBI, NIH, n.d.). It may also be caused by any injury or some disease affecting the normal working of the arteries (Coronary Heart Disease | NHLBI, NIH, n.d.). Coronary heart disease is sometimes also known as coronary artery disease (Coronary Artery Disease: Causes, Symptoms, and Treatment, n.d.) and can also lead to a heart attack (Coronary Artery Disease: Causes, Symptoms, and Treatment, n.d.). According to a survey by Centers for Disease Control and Prevention (CDC), it is the most common type of heart disease in the USA accounting for around 655,000 deaths per year (Coronary Artery Disease: Causes, Symptoms, and Treatment, n.d.).

Therefore, there is a need of some automatic diagnosis system that can process huge amounts of collected medical data and provide help with decision making. Machine learning contains large number of tools from preprocessing of data to the prediction process which can be utilized to help medical professionals in making decisions. In recent times, machine learning classifiers have been used extensively for decision making in various fields (Ashraf, Zaman & Ahmed, 2018a, 2018b, 2019, 2020; Mir et al., 2016; Mohd, Butt & Baba, 2019, 2020; Zaman, Quadri & Butt, 2012; Zaman, Kaul & Ahmed, 2020) as well as in medical science including diabetes (Kavakiotis et al., 2017; Wei, Zhao & Miao, 2018) liver diseases (El-Shafeiy, El-Desouky & Elghamrawy, 2018; Wu et al., 2019), brain diseases (Sakai & Yamada, 2019; Salvatore et al., 2014) and heart diseases. In this study various machine learning base and ensemble classifiers have been used for coronary

heart disease prediction. In the recent past, various machine learning classifiers that have been used for heart disease prediction include logistic regression, artificial neural networks, random forest, naïve bayes, k-nearest neighbor etc. In this study, the performance of various machine learning base and ensemble classifiers has been evaluated. The dataset used for this purpose is obtained from the Framingham Heart Study which is a long-term, ongoing cardiovascular study of people from Framingham city in Massachusetts (Framingham Heart Study, n.d.). The said dataset contains 4240 records with 15 attributes and one target attribute. Out of 4240 rows, there were 582 rows with missing values that were handled by removing them from our dataset. Also, the dataset contained various outliers which were also handled by removing them from our dataset. Various classifiers used include k-nearest neighbor, multilayer perceptron, multinomial naïve bayes, logistic regression, decision tree, random forest and support vector machine classifiers. Ensemble classifiers used include majority voting, weighted average, bagging and boosting classifiers. The block diagram of the proposed system is shown in Fig. 5.

The rest of this paper is organized as; Section 2 shows the literature review of the past years related to early detection of various heart diseases. Section 3 enlightens various machine learning classifiers that have been used extensively in the past years for heart disease prediction. In section 4, proposed methodology has been described that includes source of the data, features description, classification, validation schemes and evaluation metrics. Section 5 is the results and discussion section and the last section (section 6) summarizes the conclusion.

## **2. LITERATURE REVIEW**

Hidayet Takci et al. proposed a new method for heart attack prediction using significant features to determine which machine learning method is best for predicting heart attack. The dataset for this purpose is obtained from the Statlog heart disease dataset. According to the results, the algorithm that came out with the best prediction accuracy was support vector machine algorithm using linear kernel while as the feature selection algorithm that gave the best results was the reliefF method. Together they achieved an overall accuracy of 84.81% (Takci, 2018).

Gokulnath et al. have proposed an optimization function based on support vector machine and significant attributes are selected using objective function in genetic algorithm. Genetic algorithm (GA) and support vector machine (SVM) results are compared with other existing feature selection methods. The proposed framework is exhibited on MATLAB using Cleveland dataset and 7 significant features have been identified by GA-SVM pair. This pair together provided an average accuracy of 84.40% (Gokulnath & Shantharajah, 2019).

Benhar et al. have proposed data preparation method which is performed before data mining process for heart disease. This is done by performing systematic mapping study. A total of 58 papers are being selected from Jan 2000 up to Dec 2017. According to the results, data preparation step has been done extensively in order to increase the prediction accuracy of different algorithms and their main focus was on reducing data and performing feature selection (Benhar, Idri & Fernández-Alemán, 2019).

Bashir et al. have focused on the feature selection methods in order to increase the heart disease prediction accuracy and instead of one, many heart disease datasets have been used for improving the accuracy. For feature selection algorithms that are being used include random forest, naïve bayes, support vector machine, logistic regression, logistic regression SVM and decision tree. According to the results, logistic regression SVM and naïve bayes have achieved the highest accuracies of 84.85% and 84.24% respectively (Bashir et al., 2019).

Chandra Shekar et al. have proposed a hybrid technique of ensemble classifier using genetic algorithm with decision tree for heart disease prediction. The output achieved from GA is the optimized feature set, which is then given as an input to decision tree algorithm to give us the final result. According to results, the proposed approach achieves an accuracy of 85.37% (Chandra Shekar, Chandra & Venugopala Rao, 2019).

Latha et al. have combined multiple weak classifiers to form their ensemble for improving the accuracy of heart disease prediction. According to the results, bagging and boosting ensemble methods improve the prediction accuracy of weak classifiers up to a large extent. An overall of 7% increase in accuracy was recorded while using ensemble methods as compared to base classifiers. Also, feature selection was also done thereby resulting in an overall accuracy of 85.48% by the proposed model (Latha & Jeeva, 2019).

Thaiparnit et al. proposed a novel method for heart disease prediction using a technique known as Hoeffding Tree. The data for this purpose is obtained from UCI Repository containing 199 records with 13 attributes. Vertical Hoeffding Decision Tree (VHDT) is used for analysing the data. As per the results, an improved accuracy of 85.43% is achieved by the proposed VHDT technique (Thaiparnit, Kritsanasung & Chumuang, 2019).

Shinde et al. have utilized k means and naïve bayes techniques to design an intelligent system for heart disease diagnosis. K means clustering has been used for grouping different combination of features whileas naïve bayes has been used for prediction purpose (Shinde et al., 2015). Riyaz et al. performed a quantitative review of various machine learning techniques used for heart disease prediction (Riyaz et al., 2022).

Otoom et al. have proposed a real time monitoring system utilizing wearable sensors and mobile technology for coronary artery disease detection using machine learning. As per results the system achieved an overall accuracy of 85.1% using support vector machine with feature selection (Otoom et al., 2015). Dun et al. proposed a heart disease diagnosis system using ensemble learning. Authors have applied various ensemble and deep learning techniques using the concept of hyper-parameter tuning. The model resulted in an average accuracy of 78% on the test set (Dun, Wang & Majumder, 2016).

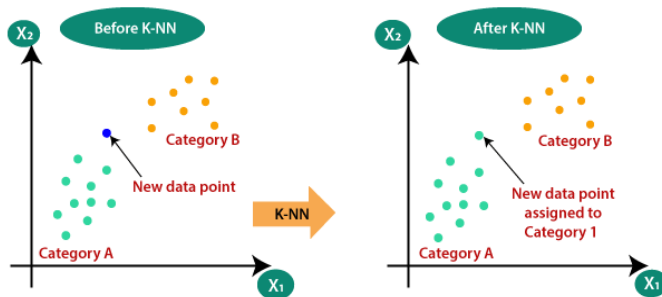
### **3. MACHINE LEARNING CLASSIFIERS**

#### **3.1. K-Nearest Neighbors**

K-Nearest Neighbor (KNN) is a simple supervised machine learning technique which compares the similarity between the new unknown data point with already existing classes. It then assigns the class to the newly arrived data point based on its resemblance with the existing classes that is most suitable for it. So whenever a new data point arrives it is being easily classified into its suitable category. Algorithm for KNN classifier is shown below:

- Step 1: Select the value for  $K$ .
- Step 2: Compute Euclidean distance for all points.
- Step 3: Depending upon distance, select the  $K$  nearest neighbors.
- Step 4: Calculate the number of data points for each category of  $K$  neighbors
- Step 5: Assign the newly arrived point to that category whose data points are more in number in  $K$  nearest neighbors.
- Step 6: Model is ready (K-Nearest Neighbor(KNN) Algorithm for Machine Learning – Javatpoint, n.d.).

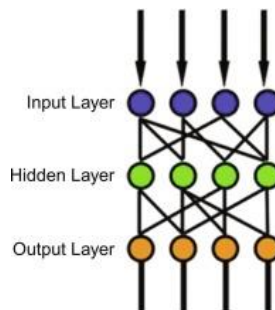
Fig. 1 shows the pictorial representation of working of KNN.



**Fig. 1. Working of KNN (K-Nearest Neighbor(KNN) Algorithm for Machine Learning – Javatpoint, n.d.)**

### 3.2. Multilayer Perceptron

Multi-layer perceptron (MLP) is an add-on of feed forward neural network. It consists of three types of layers – the input layer, hidden layer and an output layer. Input signal is fed to the input layer which processes it. The job of output layer is to perform prediction and classification. In between input and output layers, there exist one or more hidden layers considered computational engine for multilayer perceptron. The flow of data is from input to the output layer. Neurons are trained using back propagation learning algorithm (Multilayer Perceptron – an overview | ScienceDirect Topics, n.d.). Fig. 2 displays different layers of the multilayer perceptron.



**Fig. 2. Multilayer Perceptron Layers (Multilayer Perceptron – an Overview | ScienceDirect Topics, n.d.)**

The computations taking place at every neuron in the output and hidden layer are as follows:

$$o(x) = G(b(2) + W(2)h(x)) \quad (1)$$

$$h(x) = \phi(x) = s(b(1) + W(1)x) \quad (2)$$

where  $b(1)$  and  $b(2)$  are bias vectors;  $W(1)$  and  $W(2)$  as weight matrices and  $G$  and  $s$  as activation functions. The set of parameters to learn is the set  $\theta = \{W(1), b(1), W(2), b(2)\}$ . Typical choices for  $s$  include tanh function with  $\tanh(a) = (e^a - e^{-a})/(e^a + e^{-a})$  or the logistic sigmoid function, with  $\text{sigmoid}(a) = 1/(1 + e^{-a})$  (Multilayer Perceptron – an overview | ScienceDirect Topics, n.d.).

### 3.3. Multinomial Naïve Bayes

Multinomial Naive Bayes algorithm is a probabilistic learning algorithm being used for natural language processing but can also be used for other purposes such as classification. Multinomial Naïve Bayes algorithm is based on the Bayes theorem. Naive Bayes algorithm is a collection of many algorithms based on the principle that each feature being classified is not related in any way to other feature. The presence or absence of one feature does not affect the presence or absence of another feature (Multinomial Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2021 | UpGrad Blog, n.d.).

Bayes theorem (given by Thomas Bayes) is based on the formula:

$$P(A|B) = P(A) * \frac{P(B|A)}{P(B)} \quad (3)$$

read as “probability of class A given B” (Multinomial Naive Bayes Explained Function, Advantages & Disadvantages, Applications in 2021 | UpGrad Blog, n.d.), where  $P(B)$  is the prior probability of B,  $P(A)$  is prior probability of class A, and  $P(B|A)$  is the occurrence of predictor B given class A probability (Multinomial Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2021 | UpGrad Blog, n.d.).

### 3.4. Logistic Regression

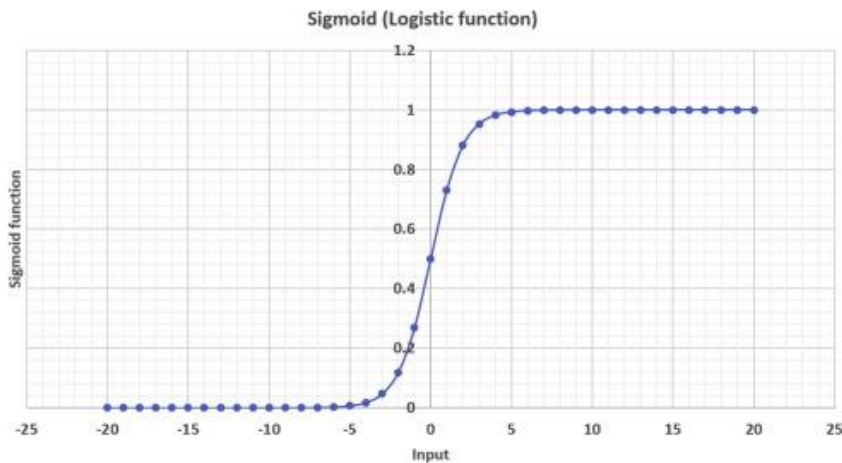
Logistic regression is another powerful machine learning algorithm based on supervised learning. It is mostly used for binary classification problems. In other words logistic regression is a linear regression used for classification task (Logistic Regression – an overview | ScienceDirect Topics, n.d.).

It normally uses a logistic function to model a binary output variable.

$$\text{Logistic function} = 1/(1 + e^{-x}) \quad (4)$$

where  $x$  is the input variable (Logistic Regression – an overview | ScienceDirect Topics, n.d.). The difference between linear regression and logistic regression is that the range of logistic regression is limited to between 0 and 1 only (Logistic Regression – an overview | ScienceDirect Topics, n.d.). In addition to this there is no need of linear relationship between input variables and output variables as far as logistic regression is concerned (Logistic Regression – an overview | ScienceDirect Topics, n.d.).

The following Fig. 3 shows logistic regression applied to the range -20 to 20 (Logistic Regression – an overview | ScienceDirect Topics, n.d.).



**Fig. 3. Logistic regression applied to the range (-20, 20)**  
(Logistic Regression – an overview | ScienceDirect Topics, n.d.)

### 3.5. Decision Tree

Decision tree is the most powerful and popular tool for classification and prediction problems (Decision Tree – GeeksforGeeks, n.d.). It is a tree-like structure in which each internal node represents an attribute test and branches designate test outcomes. The leaf nodes represent the class label (Decision Tree – GeeksforGeeks, n.d.).

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making (Decision Trees in Machine Learning | by Prashant Gupta | Towards Data Science, n.d.). It uses a tree-like structure of decisions. In addition to being extensively used in data mining, it has also been extensively used in machine learning for making decisions (Decision Trees in Machine Learning | by Prashant Gupta | Towards Data Science, n.d.).

Following is the top-down approach of the tree construction algorithm (Data Jabberwocky: Decision Tree Mathematical Formulation, n.d.):

1. A search method to compose the tree.
2. Node splitter – to split nodes.
3. Stop criterion – rule to stop the process.
4. Split acceptor – the rule that decides either to accept the best split of a node or to make it a leaf.
5. Split prospects estimator – procedure that determines order of the nodes.
6. Split – order in which nodes need to be split.
7. Decision-making module – provides decisions for data items.
8. Optional data transformations – performs preprocessing of data (Data Jabberwocky: Decision Tree Mathematical Formulation, n.d.).

Entropy and information gain are used when dealing with decision tree. Entropy ( $E$ ) is a way of measuring how mixed a column is (Entropy and Information Gain in Decision Trees | by Jeremiah Lutes | Towards Data Science, n.d.). Specifically, it is used to measure



the disorder while as information gain ( $IG$ ) is used to determine the best features/attributes for split in a decision tree. Formulas for entropy and gain are as under (Entropy: How Decision Trees Make Decisions | by Sam T | Towards Data Science, n.d.):

$$E(S) = \sum_{i=1}^c -p_i \log_2 p_i \quad (5)$$

$$IG(Y, X) = E(Y) - E\left(\frac{Y}{X}\right) \quad (6)$$

## 4. METHODOLOGY

### 4.1. Data Source

For this study, the dataset was obtained from the Framingham Heart Study which is a long-term, ongoing cardiovascular study of people from Framingham city in Massachusetts, US to estimate the ten year risk of developing coronary heart disease in order to assess the 10-year cardiovascular disease risk. It started with 5209 adult participants in its first generation and is currently in its third generation. It is carried out by National Heart, Lung, and Blood Institute staffed by various Boston University professionals. Cerebrovascular events, peripheral artery disease and heart failure were subsequently added as disease outcomes for the 2008 framingham risk score on top of coronary heart disease. The said dataset contains 4240 records with 15 attributes and one target attribute (Framingham Heart Study, n.d.). Fig. 4 shows the matrix of correlations among various attributes of the dataset that shows how the attributes of the dataset are related to each other and with the target attribute.

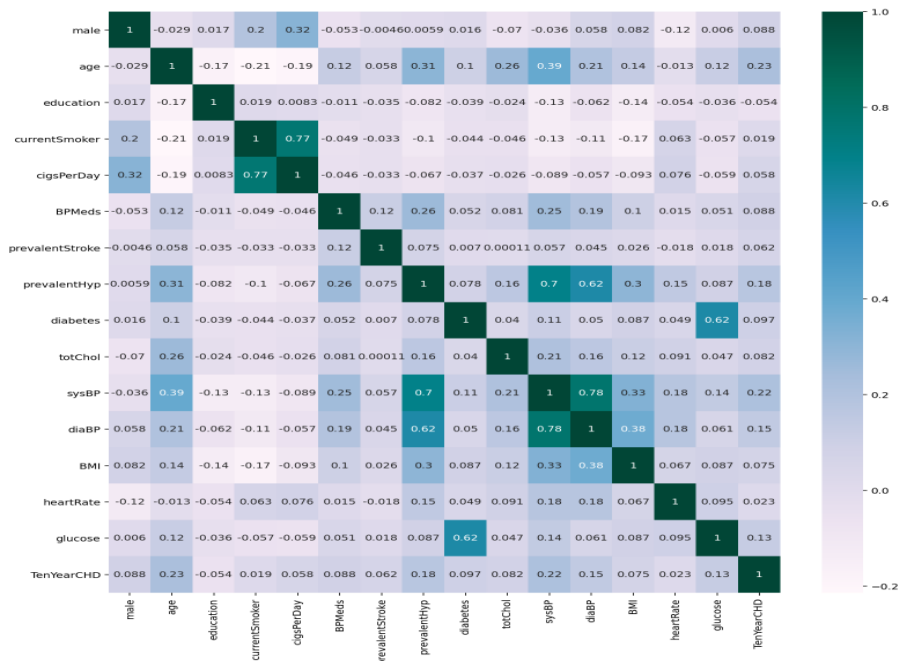


Fig. 4. Correlation Matrix

Tab. 1 shows the names and description of all 16 attributes.

**Tab. 1. List of attributes along with their description**

S. No.	Attributes	Description	Data type
1.	<i>Male</i>	Gender male or female	Nominal: “1” means “male”, “0” means “female”
2.	<i>Age</i>	Age of the patient	Continuous
3.	<i>Education</i>	Education of patient	Categorical
4.	<i>Current Smoker</i>	Whether or not the patient is a current smoker	Nominal
5.	<i>CigsPerDay</i>	Number of cigarettes smoked per day	Continuous
6.	<i>BPMeds</i>	Whether or not patient was on blood pressure medication	Nominal
7.	<i>Prevalent Stroke</i>	Whether patient had any stroke previously	Nominal
8.	<i>PrevalentHyp</i>	Whether or not the patient was hypertensive	Nominal
9.	<i>Diabetes</i>	Whether or not the patient had diabetes	Nominal
10.	<i>TotChol</i>	Total cholesterol level	Continuous
11.	<i>SysBP</i>	Systolic blood pressure	Continuous
12.	<i>DiasBP</i>	Diastolic blood pressure	Continuous
13.	<i>BMI</i>	Body Mass Index	Continuous
14.	<i>Heart Rate</i>	Heart rate of patient	Continuous
15.	<i>Glucose</i>	Glucose level of patient	Continuous
16.	<i>TenYearCHD</i> (Target variable)	Whether the patient has 10-year future risk of Coronary Heart Disease (CHD)	Binary: “1”, means “Yes”, “0” means “No”

As shown in Tab. 1, there were a total of 16 attributes: *Male* – the gender of the patient. It is a binary variable named ‘male’ in the dataset, “1” means “male” and “0” means “female”. *Age* – age of the patient (in years) at medical examination time. *Education* – a categorical variable about patient’s education (“1” means “some high school”, “2” means “high school/GED”, “3” means “some college/vocational school”, and “4” means “college”). *CurrentSmoker* – is the patient current smoker at the time of examination. *CigsPerDay* – number of cigarettes smoking each day. *BPMeds* – whether using any anti-hypertensive medication at examination. *PrevalentStroke* – whether any prevalent stroke (“0” means free of disease). *PrevalentHyp* – whether prevalent hypertensive. *Diabetes* – whether diabetic. *TotChol* – total cholesterol in mg/dL. *SysBP* – systolic blood pressure in mmHg. *DiasBP* – diastolic blood pressure in mmHg. *BMI* – body mass index (weight in kg/ height in m<sup>2</sup>).

*HeartRate* – heart rate in beats/minute. *Glucose* – glucose level in mg/dL and finally the target variable *TenYearCHD* – whether risk of ten year coronary heart disease in future (“1” means “Yes” and “0” means “No”).

## 4.2. Validation Schemes and Evaluation Metrics

### 4.2.1. Validation Schemes

In this study, experiments were performed using the train-test-split scheme. The 70–30 train-test data partitioning scheme was used. That is, the dataset was split into two parts using 70–30% split. Therefore, the classifiers were trained using 2561 records of patients in the first experiment and 2228 records in the next one and tested using remaining 1097 and 955 records for first and second (without outliers) experiment respectively.

### 4.2.2. Evaluation Metrics

To evaluate the performance of classifiers, different evaluation metrics including accuracy, precision, recall and f1-score were used. *Accuracy* is the percentage of correctly classified subjects in the test dataset. *Recall* conveys information about the percentage of correctly classified subjects while *precision* conveys information about correctly classified healthy subjects. The formulation of these evaluation metrics is given as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

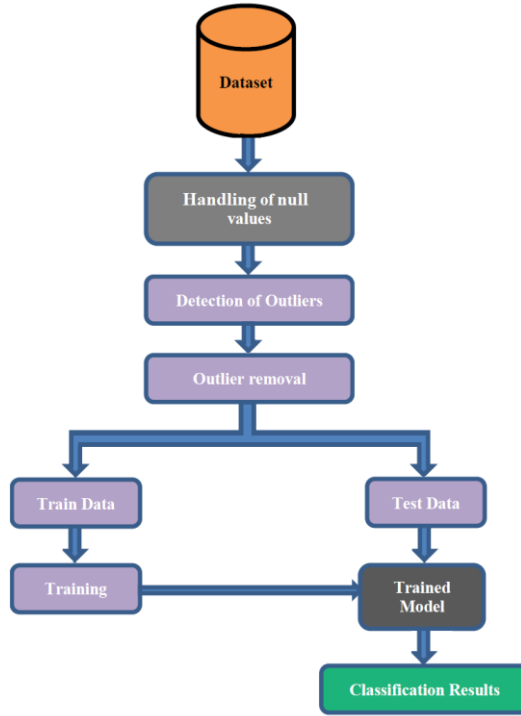
where: *TP* denotes number of true positives, *FP* denotes number of false sitives, *TN* denotes number of true negatives and *FN* denotes number of false negatives.

$$Recall = TP / (TP + FN) \quad (8)$$

$$Precision = TN / (TN + FP) \quad (9)$$

## 4.3. Classification

Fig. 5 shows the block diagram of the proposed system. Firstly, preprocessing of the aforementioned dataset was done. The dataset contained 4240 records with 15 attributes and one target attribute. Out of 4240 rows, there were 582 rows with null values. The columns that contained missing values were *education*, *cigsPerDay*, *totChol*, *heartRate*, *BPMeds*, *BMI* and *glucose*. The number wise null values in each column were as: *education* (105 null values), *cigsPerDay* (29 null values), *totChol* (50 null values), *heartRate* (1 null value), *BPMeds* (53 null values), *BMI* (19 null values) and *glucose* (388 null values). The number and percentage wise null values in each column are shown in Tab. 2.



**Fig. 5. Block diagram of the proposed system**

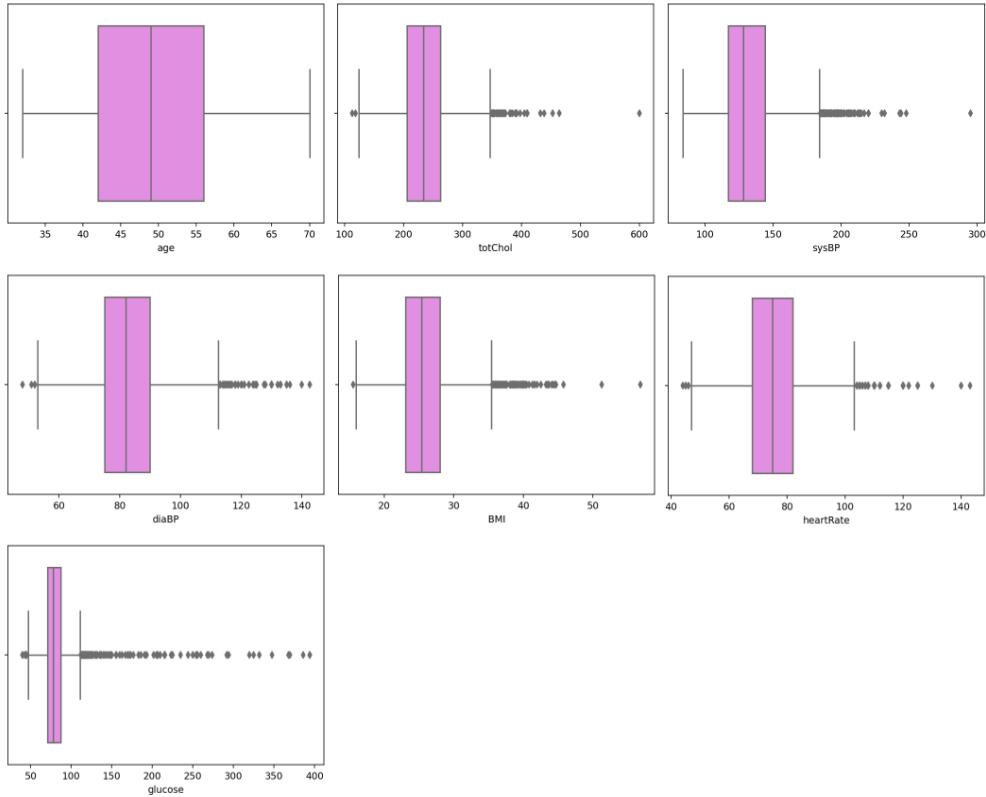
Null values in a dataset can cause hindrances for the classifiers during the prediction process. Therefore, in this study null values were handled by removing all the rows with null values from the dataset.

**Tab. 2. Percentage of missing values for each attribute**

Attribute	Number of missing values	Percentage of missing values (in %)
<i>Male</i>	0	0.00
<i>Age</i>	0	0.00
<i>Education</i>	105	2.48
<i>Current Smoker</i>	0	0.00
<i>Cigarettes per day</i>	29	0.68
<i>BPMeds</i>	53	1.25
<i>Prevalent stroke</i>	0	0.00
<i>Prevalent hypertensive</i>	0	0.00
<i>Diabetes</i>	0	0.00
<i>Total cholesterol</i>	50	1.18
<i>SysBP</i>	0	0.00
<i>DiasBP</i>	0	0.00
<i>BMI</i>	19	0.45
<i>Heart rate</i>	1	0.02
<i>Glucose</i>	388	9.15

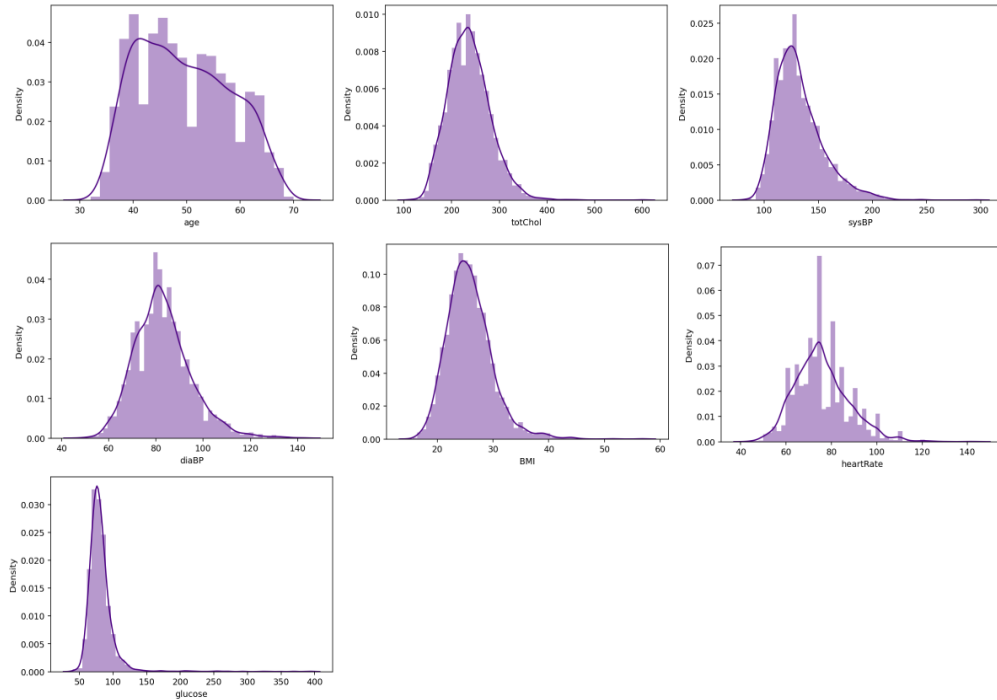
After removing all the rows with null values, our dataset was now left with 3658 records of patients only.

In the next step, the task of outlier detection was performed. Out of all attributes, *age*, *totChol*, *sysBP*, *diaBP*, *BMI*, *heartRate* and *glucose* were the only attributes with continuous values. Therefore the outliers were detected for these columns only. The figure (Fig. 6) shows the plot representation of outliers for these attributes.



**Fig. 6. Plot showing outliers**

As seen in the figure, attributes *totChol*, *sysBP*, *diaBP*, *BMI*, *heartRate* and *glucose* contained outliers. Fig. 7 shows the distribution of values for these columns.



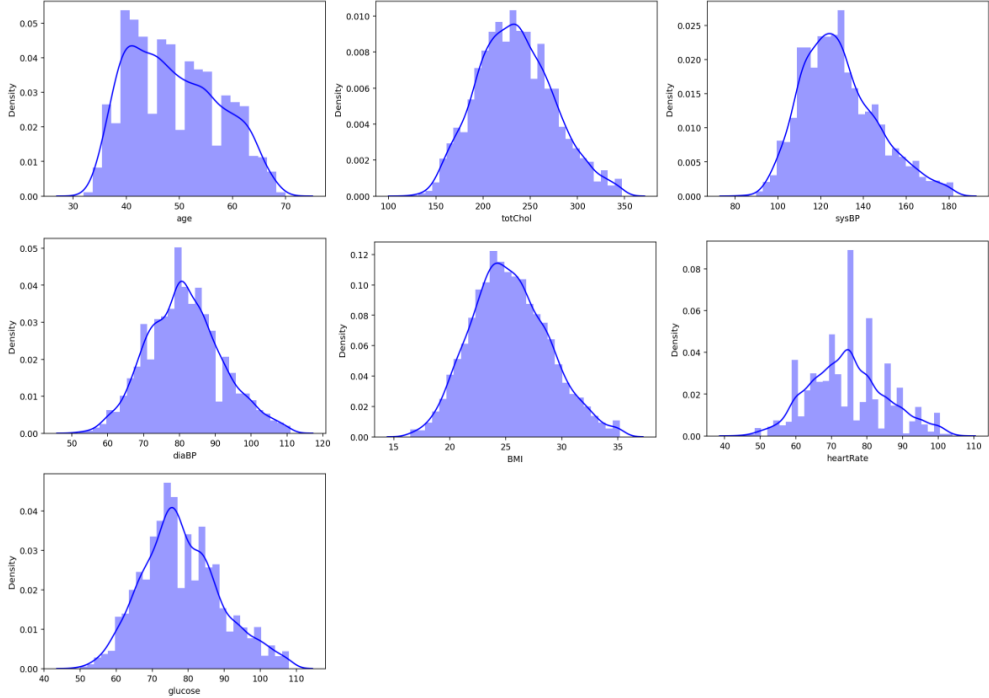
**Fig. 7. Plot showing distributions for the Continuous values**

The number of outliers in each of these columns was as; *totChol* (46 outliers), *sysBP* (116 outliers), *diaBP* (32 outliers), *BMI* (66 outliers), *heartRate* (60 outliers) and *glucose* (155 outliers) ( Tab. 3). Therefore, the outliers were handled by removing all the outliers from *totChol*, *sysBP*, *diaBP*, *BMI*, *heartRate* and *glucose* columns.

**Tab. 3. Number of outliers present**

Attribute	Number of Outliers
totChol	46
sysBP	116
diaBP	32
BMI	66
heartRate	60
glucose	155

After removing outliers from our data, the number of instances in our data got reduced from 3658 to 3183 records. Fig. 8 shows the distribution of continuous values after outlier treatment for these columns.



**Fig. 8. Plot showing distributions after Outlier Treatment**

After treatment of outliers, the distribution looked normal (as shown in Fig. 8). After preprocessing of data, the dataset was then divided into 2 parts: training set and test set where 70% of the data was used for training purpose and remaining 30% was used for testing. In this study, no feature selection technique was used therefore the whole feature set was used for training and testing of classifiers. The machine learning classifiers that were used for this study include: k-nearest neighbor, multi-layer perceptron, support vector machine, multinomial naïve bayes, logistic regression, decision tree and random forest algorithms. Tab. 4 shows the performance metrics of each of these machine learning classifiers.

The same data was then used for training ensemble classifiers. Tab. 5 shows the performance metrics of these classifiers after outlier treatment.

## 5. RESULTS AND DISCUSSION

To evaluate the performance of the classifiers, the pre-processed data was supplied to all the aforementioned classifiers in order to calculate their efficiencies. All the computations were done on Intel Core I3 processor on windows 10 operating system. Code was written in *python* language using Anaconda3 *Spyder* platform.

In the first experiment, 2561 records were used to train all the mentioned 13 machine learning classifiers without removal of outliers and were then tested using the remaining 1097 records to evaluate their efficiencies. Tab. 4 shows the performance achieved by various classifiers without outlier treatment.

**Tab. 4. Performance achieved by various classifiers without outlier treatment**

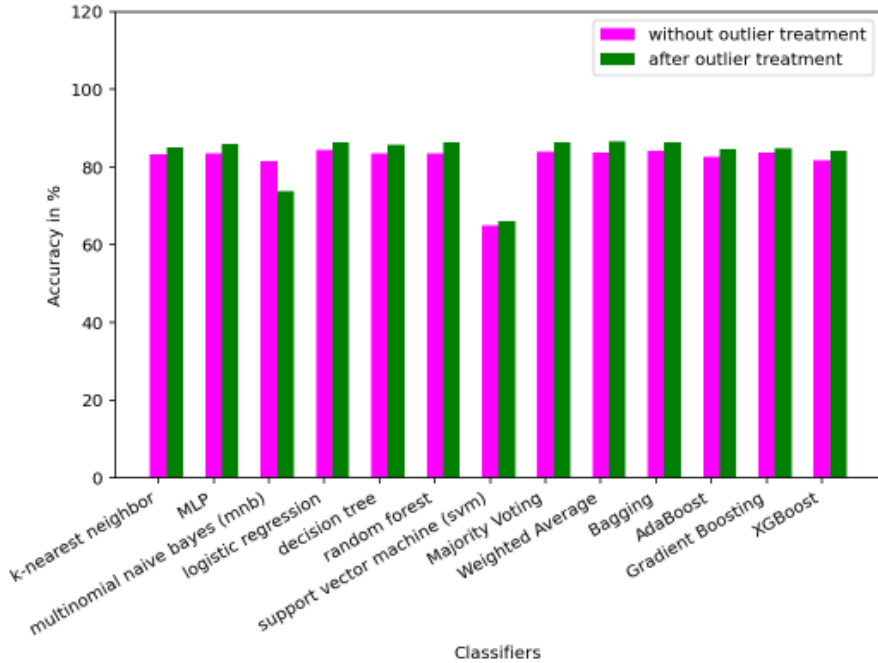
Classifier	Performance Metrics			
	Precision	Recall	F1 Score	Accuracy
<i>K Nearest Neighbor (KNN)</i>	0.78	0.83	0.79	0.8324
<i>Multilayer Perceptron (MLP)</i>	0.70	0.84	0.76	0.8361
<i>Multinomial Naïve Bayes (MNB)</i>	0.78	0.82	0.79	0.8151
<i>Logistic Regression (LR)</i>	0.83	0.84	0.79	0.8443
<i>Decision Tree (DT)</i>	0.77	0.84	0.77	0.8361
<i>Random Forest (RF)</i>	0.77	0.84	0.77	0.8361
<i>Support Vector Machine (SVM)</i>	0.82	0.65	0.70	0.6503
<i>Majority Voting</i>	0.80	0.84	0.78	0.8397
<i>Weighted Average</i>	0.80	0.84	0.77	0.8379
<i>Bagging</i>	0.81	0.84	0.79	0.8424
<i>AdaBoost</i>	0.78	0.83	0.79	0.8270
<i>Gradient Boosting</i>	0.79	0.84	0.78	0.8370
<i>XGBoost</i>	0.76	0.82	0.78	0.8179

**Tab. 5. Performance achieved by classifiers after outlier treatment**

Classifier	Performance Metrics			
	Precision	Recall	F1 Score	Accuracy
<i>K Nearest Neighbor (KNN)</i>	0.78	0.85	0.80	0.8503
<i>Multilayer Perceptron (MLP)</i>	0.79	0.86	0.80	0.8597
<i>Multinomial Naïve Bayes (MNB)</i>	0.78	0.74	0.76	0.7382
<i>Logistic Regression (LR)</i>	0.82	0.86	0.80	0.8639
<i>Decision Tree (DT)</i>	0.77	0.86	0.80	0.8565
<i>Random Forest (RF)</i>	0.81	0.86	0.81	0.8628
<i>Support Vector Machine (SVM)</i>	0.84	0.66	0.71	0.6597
<i>Majority Voting</i>	0.80	0.86	0.80	0.8628
<i>Weighted Average</i>	0.88	0.86	0.80	0.8649
<i>Bagging</i>	0.82	0.86	0.81	0.8639
<i>AdaBoost</i>	0.79	0.85	0.81	0.8450
<i>Gradient Boosting</i>	0.79	0.85	0.81	0.8482
<i>XGBoost</i>	0.78	0.84	0.80	0.8419

Then in next experiment, the same set of classifiers was again trained using the training set but this time the outliers were first removed from the data. A total of 475 rows were detected that contained null values and after being removed, the dataset was left with 3183 records of patients only. Therefore in the second experiment, only 2228 instances (70%) of patients were used for training of classifiers and remaining 955 (30%) records were used for evaluating their performance. Tab. 5 shows the performance achieved by various machine learning classifiers after outlier treatment. Fig. 9 shows performance comparison of all classifiers with and without outlier treatment using bar-plot (Fig. 9).





**Fig. 9. Bar-plot showing differences in accuracies of classifiers with and without outlier treatment**

As shown in the figure (Fig. 9), it can be clearly seen that there was a difference in the accuracies achieved by each of these machine learning classifiers with and without the outlier treatment excluding the one that is multinomial naïve bayes classifier for which accuracy got decreased upon removal of outliers. All the rest of the classifiers showed an accuracy improvement when outliers were removed from the data.

With removal of outliers, the increase in the percentage of accuracies for each of these classifiers was as k-nearest neighbor (1.79%), multilayer perceptron (2.36%), logistic regression (1.96%), decision tree (2.04%), random forest (2.67%), support vector machine (0.94%), majority voting (2.31%), weighted average (2.70%), bagging (2.15%), adaboost (1.80%), gradient boosting (1.12%) and xgboost (2.4%) respectively. Therefore, it was concluded that outliers in data affect the overall efficiency of the machine learning classifiers.

## 6. CONCLUSION

Heart is a vital organ of the human body performing the crucial function of pumping blood to different body parts. A slight change in the normal functioning of the heart can lead to imbalance in the functioning of the whole body. Heart diseases are the main reason behind deaths in the world today. So, detection of heart disease at an early stage can help in saving numerous human lives. Heart disease is a general term for various types of heart conditions and one such kind is the coronary heart disease. Coronary heart disease is caused by blockage of heart arteries by fat deposits around heart. The aim of this paper is prediction of coronary heart disease using machine learning. Various machine learning base (such as k-nearest neighbor, multilayer perceptron, multinomial naïve bayes, logistic

regression, decision tree, random forest, support vector machine) and ensemble classifiers have been used in this study. The dataset was obtained from the Framingham Heart Study database comprising 4240 instances with 15 attributes for each instance. Firstly, the data was preprocessed by removing all the rows containing null values from the data. Then in the next step the data was checked for presence of any outliers and hence removed accordingly. Then finally in the last step, various machine learning base and ensemble classifiers were trained and tested using the given dataset for predicting the coronary heart disease first including outliers and then in the second experiment without outliers. As per our results, classifiers performed better in the second experiment where the outliers were first removed from the data as compared to the previous experiment where the outliers were also included. In addition to this, the classifier that came out with the best performance among all was weighted average ensemble classifier achieving an accuracy of almost 86.50% in the second experiment. Therefore, it was concluded that outliers in data affect the overall efficiency of the machine learning classifiers.

In future, accuracy can further be improved by reducing the number of features under consideration.

## Acknowledgments

*The authors of this paper would like to extend their gratitude to all those people/organizations that provided their expertise and assistance at different points during the course of this study.*

## Conflicts of Interest

*The authors have no conflicts of interest to declare.*

## REFERENCES

- Ashraf, M., Zaman, M., & Ahmed, M. (2018a). Using ensemble stacking method and base classifiers to ameliorate prediction accuracy of pedagogical data. *Procedia Computer Science*, 132(Iccids), 1021–1040. <https://doi.org/10.1016/j.procs.2018.05.018>
- Ashraf, M., Zaman, M., & Ahmed, M. (2018b). Performance analysis and different subject combinations: an empirical and analytical discourse of educational data mining. *Proceedings of the 8th International Conference Confluence 2018 on Cloud Computing, Data Science and Engineering, Confluence 2018* (pp. 287–292). IEEE. <https://doi.org/10.1109/CONFLUENCE.2018.8442633>
- Ashraf, M., Zaman, M., & Ahmed, M. (2019). To ameliorate classification accuracy using ensemble vote approach and base classifiers. In *Advances in Intelligent Systems and Computing* (vol 813). Springer Singapore. [https://doi.org/10.1007/978-981-13-1498-8\\_29](https://doi.org/10.1007/978-981-13-1498-8_29)
- Ashraf, M., Zaman, M., & Ahmed, M. (2020). An intelligent prediction system for educational data mining based on ensemble and filtering approaches. *Procedia Computer Science*, 167(2019), 1471–1483. <https://doi.org/10.1016/j.procs.2020.03.358>
- Bashir, S., Khan, Z. S., Hassan Khan, F., Anjum, A., & Bashir, K. (2019). Improving Heart Disease Prediction Using Feature Selection Approaches. *Proceedings of 2019 16th International Bhurban Conference on Applied Sciences and Technology*, (pp. 619–623). IEEE. <https://doi.org/10.1109/IBCAST.2019.8667106>
- Benhar, H., Idri, A., & Fernández-Alemán, J. L. (2019). A Systematic Mapping Study of Data Preparation in Heart Disease Knowledge Discovery. *Journal of Medical Systems*, 43(1), 17. <https://doi.org/10.1007/s10916-018-1134-z>

- Cardiovascular (Heart) Diseases: Types and Treatments. (n.d.). Retrieved January 8, 2022 from <https://www.webmd.com/heart-disease/guide/diseases-cardiovascular>
- Chandra Shekar, K., Chandra, P., & Venugopala Rao, K. (2019). An Ensemble Classifier Characterized by Genetic Algorithm with Decision Tree for the Prophecy of Heart Disease. In *Lecture Notes in Networks and Systems* (Vol. 74). Springer Singapore. [https://doi.org/10.1007/978-981-13-7082-3\\_2](https://doi.org/10.1007/978-981-13-7082-3_2)
- Coronary artery disease: Causes, symptoms, and treatment. (n.d.). Retrieved December 22, 2021 from <https://www.medicalnewstoday.com/articles/184130>
- Coronary heart disease – NHS. (n.d.). Retrieved December 22, 2021 from <https://www.nhs.uk/conditions/coronary-heart-disease/>
- Coronary Heart Disease | NHLBI, NIH. (n.d.). Retrieved December 22, 2021 from <https://www.nhlbi.nih.gov/health-topics/coronary-heart-disease>
- Data Jabberwocky: Decision Tree Mathematical Formulation. (n.d.). Retrieved December 26, 2021 from <http://fiascodata.blogspot.com/2018/08/decision-tree-mathematical-formulation.html>
- Decision Tree – GeeksforGeeks. (n.d.). Retrieved December 26, 2021 from <https://www.geeksforgeeks.org/decision-tree/>
- Decision Trees in Machine Learning | by Prashant Gupta | Towards Data Science. (n.d.). Retrieved December 26, 2021 from <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052>
- Dun, B., Wang, E., & Majumder, S. (2016). Heart Disease Diagnosis on Medical Data Using Ensemble Learning. *Computer Science*, 1(1), 1–5.
- El-Shafeiy, E. A., El-Desouky, A. I., & Elghamrawy, S. M. (2018). Prediction of Liver Diseases Based on Machine Learning Technique for Big Data. *Advances in Intelligent Systems and Computing*, 723, 362–374. [https://doi.org/10.1007/978-3-319-74690-6\\_36](https://doi.org/10.1007/978-3-319-74690-6_36)
- Entropy: How Decision Trees Make Decisions | by Sam T | Towards Data Science. (n.d.). Retrieved December 26, 2021 from <https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8>
- Entropy and Information Gain in Decision Trees | by Jeremiah Lutes | Towards Data Science. (n.d.). Retrieved December 26, 2021 from <https://towardsdatascience.com/entropy-and-information-gain-in-decision-trees-c7db67a3a293>
- Framingham Heart Study. (n.d.). Retrieved September 9, 2021 from <https://framinghamheartstudy.org/>
- Gokulnath, C. B., & Shantharajah, S. P. (2019). An optimized feature selection based on genetic approach and support vector machine for heart disease. *Cluster Computing*, 22(s6), 14777–14787. <https://doi.org/10.1007/s10586-018-2416-4>
- Heart disease – Symptoms and causes - Mayo Clinic. (n.d.). Retrieved January 8, 2022 from <https://www.mayoclinic.org/diseases-conditions/heart-disease/symptoms-causes/syc-20353118>
- K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint. (n.d.). Retrieved December 26, 2021 from <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
- Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I., & Chouvarda, I. (2017). Machine Learning and Data Mining Methods in Diabetes Research. *Computational and Structural Biotechnology Journal*, 15, 104–116. <https://doi.org/10.1016/J.CSBJ.2016.12.005>
- Latha, C. B. C., & Jeeva, S. C. (2019). Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. *Informatics in Medicine Unlocked*, 16, 100203. <https://doi.org/10.1016/j.imu.2019.100203>
- Less than \$1: How WHO thinks that can save 7 million lives. (n.d.). Retrieved January 9, 2022 from <https://www.downtoearth.org.in/news/health/less-than-1-how-who-thinks-that-can-save-7-million-lives-80679>
- Logistic Regression - an overview | ScienceDirect Topics. (n.d.). Retrieved December 26, 2021 from <https://www.sciencedirect.com/topics/computer-science/logistic-regression>
- Mir, N. M., Khan, S., Butt, M. A., & Zaman, M. (2016). An experimental evaluation of Bayesian classifiers applied to intrusion detection. *Indian Journal of Science and Technology*, 9(12), 1–13. <https://doi.org/10.17485/ijst/2016/v9i12/86291>
- Mohd, R., Butt, M. A., & Baba, M. Z. (2020). GWLM–NARX: Grey Wolf Levenberg–Marquardt-based neural network for rainfall prediction. *Data Technologies and Applications*, 54(1), 85–102. <https://doi.org/10.1108/DTA-08-2019-0130>
- Mohd, R., Butt, M. A., & Baba, M. Z. (2019). SALM-NARX: Self adaptive LM-based NARX model for the prediction of rainfall. *Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018* (pp. 580–585). IEEE. <https://doi.org/10.1109/I-SMAC.2018.8653747>

- Multilayer Perceptron – an overview | ScienceDirect Topics. (n.d.). Retrieved December 26, 2021 from <https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron>
- Multinomial Naive Bayes Explained: Function, Advantages & Disadvantages, Applications in 2021 | upGrad blog. (n.d.). Retrieved December 26, 2021 from <https://www.upgrad.com/blog/multinomial-naive-bayes-explained/>
- Otoom, A. F., Abdallah, E. E., Kilani, Y., & Kefaye, A. (2015). Effective Diagnosis and Monitoring of Heart Disease. *International Journal of Software Engineering and Its Applications*, 9(1), 143–156.
- Riyaz, L., Butt, M. A., Zaman, M., & Ayob, O. (2022). Heart Disease Prediction Using Machine Learning Techniques: A Quantitative Review. *Advances in Intelligent Systems and Computing* (pp. 81–94). Springer. [https://doi.org/10.1007/978-981-16-3071-2\\_8](https://doi.org/10.1007/978-981-16-3071-2_8)
- Sakai, K., & Yamada, K. (2019). Machine learning studies on major brain diseases: 5-year trends of 2014–2018. *Japanese Journal of Radiology*, 37, 34–72. <https://doi.org/10.1007/s11604-018-0794-4>
- Salvatore, C., Cerasa, A., Castiglioni, I., Gallivanone, F., Augimeri, A., Lopez, M., Arabia, G., Morelli, M., Gilardi, M. C., & Quattrone, A. (2014). Machine learning on brain MRI data for differential diagnosis of Parkinson's disease and Progressive Supranuclear Palsy. *Journal of Neuroscience Methods*, 222, 230–237. <https://doi.org/10.1016/J.JNEUMETH.2013.11.016>
- Shinde, R., Arjun, S., Patil, P., & Waghmare, P. J. (2015). An Intelligent Heart Disease Prediction System Using K-Means Clustering and Naïve Bayes Algorithm. *International Journal of Computer Science and Information Technolog*, 6(1), 637–639.
- Takci, H. (2018). Improvement of heart attack prediction by the feature selection methods. *Turkish Journal of Electrical Engineering and Computer Sciences*, 26(1), 1–10. <https://doi.org/10.3906/elk-1611-235>
- Thaiparnit, S., Kritsanang, S., & Chumuang, N. (2019). A Classification for Patients with Heart Disease Based on Hoeffding Tree. *JCSSE 2019 – 16th International Joint Conference on Computer Science and Software Engineering: Knowledge Evolution Towards Singularity of Man-Machine Intelligence* (pp. 352–357). IEEE. <https://doi.org/10.1109/JCSSE.2019.8864158>
- Wei, S., Zhao, X., & Miao, C. (2018). A comprehensive exploration to the machine learning techniques for diabetes identification. *IEEE World Forum on Internet of Things, WF-IoT 2018 - Proceedings*, (pp. 291–295). IEEE. <https://doi.org/10.1109/WF-IOT.2018.8355130>
- Wu, C. C., Yeh, W. C., Hsu, W. D., Islam, M. M., Nguyen, P. A., Poly, T. N., Wang, Y. C., Yang, H. C., & Li, Y. C. (2019). Prediction of fatty liver disease using machine learning algorithms. *Computer Methods and Programs in Biomedicine*, 170, 23–29. <https://doi.org/10.1016/J.CMPB.2018.12.032>
- Zaman, M., Kaul, S., & Ahmed, M. (2020). Analytical comparison between the information gain and gini index using historical geographical data. *International Journal of Advanced Computer Science and Applications*, 11(5), 429–440. <https://doi.org/10.14569/IJACSA.2020.0110557>
- Zaman, M., Quadri, S. M. K., & Butt, M. A. (2012). Information translation: A practitioners approach. *Lecture Notes in Engineering and Computer Science*, 1, 45–47.

*Keywords: source code, binary classification,  
text classification, AutoML*

*Marcin BADUROWICZ* [0000-0003-2249-4219]\*

# DETECTION OF SOURCE CODE IN INTERNET TEXTS USING AUTOMATICALLY GENERATED MACHINE LEARNING MODELS

## Abstract

*In the paper, the authors are presenting the outcome of web scraping software allowing for the automated classification of source code. The software system was prepared for a discussion forum for software developers to find fragments of source code that were published without marking them as code snippets. The analyzer software is using a Machine Learning binary classification model for differentiating between a programming language source code and highly technical text about software. The analyzer model was prepared using the AutoML subsystem without human intervention and fine-tuning and its accuracy in a described problem exceeds 95%. The analyzer based on the automatically generated model has been deployed and after the first year of continuous operation, its False Positive Rate is less than 3%. The similar process may be introduced in document management in software development process, where automatic tagging and search for code or pseudo-code may be useful for archiving purposes.*

## 1. INTRODUCTION

In the community of software development, there is a huge network of applications and websites provided for knowledge exchange. While aspiring software developers may use the community help to search for answers for their questions, the more experienced ones could respond and provide feedback but also take part in discussions about software architecture and more advanced topics. These discussions are highly technical and full of jargon, but also the requirement is to show snippets of source code in various programming languages to achieve proper comments.

Two problems are arising – to properly format the source code published on the website, it must be properly marked as a source code, using either the HTML5 code element or the triple backtick element in the Markdown system. Inability to properly mark the code will render the code snippet hard to read by other software developers, as presented in figure 1 for example. On the other hand, the misformatted code snippet may be completely impossible to understand because it lacks proper indentation, e.g., in the case of the Python programming language.

---

\* Lublin University of Technology, Faculty of Electrical Engineering and Computer Science,  
Department of Computer Science, Poland, m.badurowicz@pollub.pl

In the last decade, both Machine Learning (ML) and Natural Language Processing have been applied in the study of the source code (Ugurel, Krovetz & Giles, 2002), and the problem of classification of programming languages has been also broadly discussed (Van Dam & Zaytsev, 2016), using Neural Networks (Gilda, 2017), Bayesian learning (Khasnabish et al., 2014) or Multinomial Naïve Bayes (Alreshedy et al., 2018), Convolutional Neural Networks (CNN) (Ohashi & Watanobe, 2019) and Neural Text Classification (LeClair, Eberhart & McMillan, 2018) or even, alternatively the usage of Speech Recognition techniques (Madani et al., 2010). However, all of the presented similar works are mostly concentrating on differentiating between different programming languages than on a problem of differentiating between source code and natural text, however some of the presented concepts may be useful for the presented problem.

Two already available libraries for programming languages are also available right now, pygments for Python (Pygments – Python Syntax Highlighter, n.d.) and github-linguist (Linguist, n.d.) for Ruby, also suited for recognizing a programming language, but able to return “text” classification in case of pygments and “nil” result in Ruby when the text does not seem to be a source code.

```
#include <iostream>

int main() {

    int n = 0;
    std::cin >> n;

    std::string dummy;

    std::getline(std::cin, dummy); // ***

    for(int i = 0; i < n; ++i) {
        std::string line;
        std::getline(std::cin, line);
        std::cout << '[' << line << ']' << std::endl;
    }
}
```

**Fig. 1. Example of wrongly formatted source code on the website**

Source code understanding could be also used in other contexts, e.g., training ML systems to understand natural language to produce output in a given programming language, starting with CoNaLa dataset (Yin et al., 2018) and currently available commercial products like GitHub Copilot (GitHub Copilot – Your AI Pair Programmer, n.d.).

The problem presented in this paper is a little bit different, as there is no need to identify the programming language, only a need to differentiate between a programming language and the natural text. However, this “natural text” may include specific jargon, names of variables, or even keywords of a programming language. Additionally, the system should be able to identify programming language for any kind of given language, both with more natural-looking syntax e.g., SQL and Python, as well as the ones with a very complex or mathematical-like syntax, e.g., Prolog and Haskell.

The overall document management, especially in software development may also benefit from this kind of solution, where some parts of the pseudo-code or fragments of real life implementation may be properly detected and marked in the documentation of the software development process for easier analysis and archiving purposes.

The final problem is that the system was proposed for one of a Polish software development discussion forum (4programmers.net, 2000), where jargon and overall discussion texts are very different from that in the anglosphere. The second part of this same problem is that in case of internet communication the “natural text” may also include abbreviations, internet acronyms and emotes/emoji.

The author proposed an automated system analyzing the publications on the website and providing feedback if there is a source code block found, which was not marked properly, resulting in the wrong presentation on the website, which will be tested on the website and deployed if successful, working under a set of conditions:

- It will be automatically tracking new threads and discussions on the website,
- It should allow recognition if a single line of text is a programming language snippet or just a text,
- Text may include small fragments of programming syntax,
- Code may be in any kind of programming language,
- Text will be mostly in Polish, but may include English loanwords and jargon,
- After final deployment, if there will be an unmarked code fragment recognized, the system should notify the original author and/or moderators of the website.

## 2. THE PROPOSED SOLUTION

To solve the first problem, to recognize if the given fragment of a text is a source code or not, a few solutions were considered:

- Using an database for keywords of different programming languages, and a scoring system,
- A scoring system based on lack of whitespace characters between words and a huge number of characters normally not used in a text (e.g., square brackets, semicolons at the end of the line, and similar),
- A statistical system using the average metrics of natural language (length of the word, length of the sentence) in contrast,
- Naïve Bayes system trained on keywords of different programming languages,
- A Machine Learning-based binary classification for text.

After reviewing the similar papers mentioned earlier and similar discussions on other software development websites, the author decided to try to use the Machine Learning approach.

The author has added one more requirement – it would be best if the model could be generated automatically, without user intervention, without manual planning of Artificial Neural Network (ANN) structure (Kulisz et al., 2021) and fine-tuning (Sobaszek, Gola & Kozłowski, 2020).

Such an approach forced to use the automated machine learning (AutoML) solutions, where, provided with only dataset and the definition of the problem (in the case of this paper – text binary classification), the system can automatically generate the ML model to be used in created software, to reduce time and complexity of crafting the solution (He, Zhao & Chu, 2021).

The multi-label classification of the text has been discussed (Wever et al., 2021) as a more challenging problem, but the binary classification is already available in commercially available open-source ML toolkit for the .NET platform, ML.NET (Ahmed et al., 2019). AutoML is also available for modelling and prediction problems (Machrowska et al., 2020), e.g. using LSTM method (Szabelski, Karpiński & Machrowska, 2022). In this paper the text classification will be performed, but classification of images (Kłosowski et al., 2021) is also available with such toolkits.

## **2.1. The basic and final datasets**

Because the website in question is providing the Application Programming Interface (API) for getting the content of threads and single posts in the discussion forum module, the basic dataset was prepared by downloading several newest posts from the system using created software in the C# programming language and manually categorizing them into a tab-separated (TSV) file, where each line of the file consisted of two columns – first was deciding if the line is code or text by setting 0 (not code) or 1 (code) value, the second was a single paragraph of text extracted from the discussion. The data has been manually checked to include also longer texts with fragments of the source code inside. Finally, the basic dataset was consisting of 145 examples of text and 96 examples of code, totaling 241 lines in the TSV file, about 53 KB of text. An example of the format of the dataset is presented in figure 2. The source code in the dataset was including C#, SQL, and XML languages.

The first, basic, dataset, was used for evaluation of AutoML pipeline available in ML.NET as well as already available libraries, mentioned earlier: *pygments* and *linguist*.

*Pygments* library marked 52 out of those 96 source code samples as text, wrongly classifying 54.16% of examples, marking it completely unreliable in this given task. The *linguist*'s results were also completely unsuitable for the task, as it couldn't properly classify 95 out of 96 code fragments in the basic dataset except for one line consisting of the fragment of an XML document.

It must be noted that both libraries are better suited to classify the programming language on the whole file or repository, not on a single-line code snippet. Those particularly poor results however forced the author to solve the problem using a solution crafted specifically to the problem.

The first AutoML generated model was using the *AveragePerceptronOva* trainer and reported an accuracy of 95.83%, which was already a very promising result. The preliminary trained model (on the basic dataset) was then used to provide a final dataset.

The web scrapping system has been improved to get more posts and pre-classify them using the ML model acquired in the first step. Then, the dataset labeling has been fixed manually.



```

code    text
0        Pierwszy wynik w google.
1        /** Return true if the player with "theSeed" has won after placing at (currentRow, currentCol)
0        go.anna3 minuty temuWitam,mam pytanie, ponieważ programuje w javie gre kółko krzyżyk. Mam napi
1        public boolean checkDiagonalRight(List &lt;position&gt;positions){Map&lt;Integer, Integer&gt;
0        Czyli miej więcej tak, wygląda, że, rzeczywiście, złożoność jest liniowa (nie licząc isPrime),
1        #include &lt;iostream&gt;using namespace std;int isPrime(int n) { if (n &lt;= 3) return n &gt;
0        Dzięki za odp. prześledzę jeszcze ta funkcję z SELF JOIN.Przeczytałam o funkcji OVER (dzięki z
0        W wyniku kolumna TOTAL jest poprawna, ale date i count już nie, bo nie grupuje po dacie:
1        ....date.... ...I..count..I....total....I2017-01-26 I...1..... I.....3..... I2017-01-26 I..
0        Macie może pomysł jak pogrupować to według daty bez godziny? Kombinowałam z SELECT DISTINCT, a
0        Dzięki za odpowiedzi.Może nie będę zaczynać nowego wątku, miałbym jeszcze pytanko - co polecać
0        ok, dzięki wszystkim za odzew, wiem w czym szukać, pozdrawiam :)
0        TomaszliMoon napisał(a):Wczytywanie zmiennych umieszczonych w konstruktorze powoduje, że wczytyw
1        void CarHeap::adCar (Car obj) // tutaj też wczytywane są dane bo tworzony jest obiekt tymczaso
1        void CarHeap::carEquals(Car first,Car second) // tutaj wczytywane są dane dla dwóch obiektów t
0        Do tego miała właśnie służyć funkcja readPower(). Patrzac po paru przykładach w internecie wyd
0        TomaszliMoon napisał(a):Wczytywanie zmiennych umieszczonych w konstruktorze powoduje, że wczytyw
1        void CarHeap::carEquals(Car first,Car second) // tutaj wczytywane są dane dla dwóch obiektów t
0        W tych dwóch przypadkach konstruktor się nie odpala (odpala się konstruktor kopiujący).Aczkolw
0        Wczytywanie zmiennych umieszczonych w konstruktorze powoduje, że wczytywane one są za każdym raz
0        Poza tym w tej chwili funkcja carEquals niczego nie zmienia, gdyż operuje na zmiennych tymczas
0        Funkcje analityczne byłoby moim pierwszym wyborem: COUNT(0) OVER (PARTITION BY ....) , ale sil
0        Self joinem kombinowałbym tak:
1        with helper as ( select t.data, case when t.data=s.data then 1 else 0 end total_flag from tabe
0        serio, tak nazwałby przypisanie? o_o

```

**Fig. 2. Dataset fragment example**

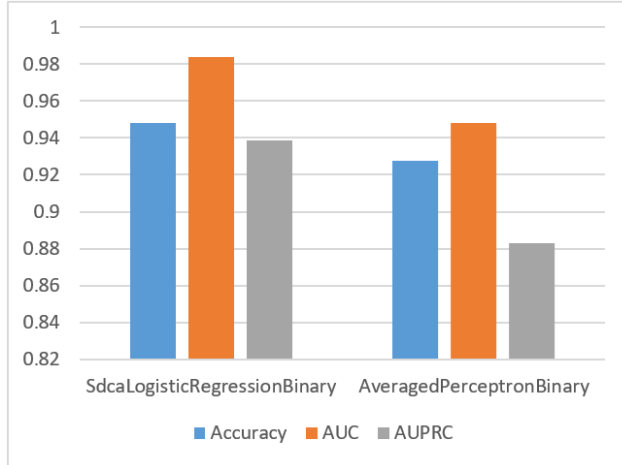
In total, the final dataset consists of 1974 lines of text (448 KB in total), consisting of 83% of regular text (which may include small fragments of code snippets) and 17% (335 lines) of pure code snippets. This dataset was including more programming languages – C#, C and C++, Java, JavaScript, HTML, XML, SQL, and Python, where all of them have their own characteristics.

## 2.2. The ML model generation using AutoML

The AutoML solution included in the ML.NET platform is based on: definition of the problem and the time given the algorithm to test for different results – during a limited time, the software is testing various algorithms, trying to find the best solution to a given problem. In the described situation, the problem is “binary-classification” and 3 runs have been tested, on Core i7-6600U machine without GPU acceleration: with 30-, 60- and 180-seconds time limits.

The ML.NET is grading the solution using three indicators: Accuracy, Area Under Precision-Recall Curve (AUPRC), and FPR Area Under the Curve (AUC).

For 30 seconds run result, the system tested two classifiers, SDCA Logistic Regression and Averaged Perceptron, both suited for binary classification problem. The indicators marking the tested solutions are presented in figure 3.



**Fig. 3. Results for 30 seconds run**

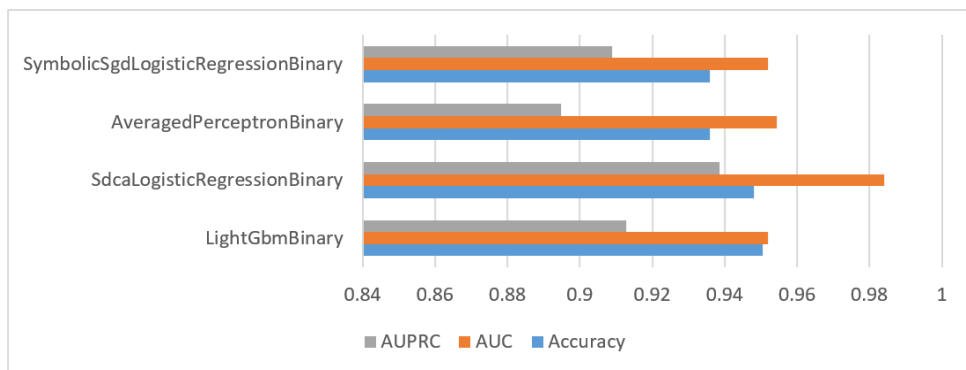
For longer runs, more algorithms have been tested, as presented in figures 4 and 5, including SVM and GBM.

Finally, the best solution returned from the ML.NET autotraining subsystem was based on SDCA Logistic Regression with an accuracy of 95.72%, AUC 95.65%, and AUPRC 88.01% returned after the 180-second run.

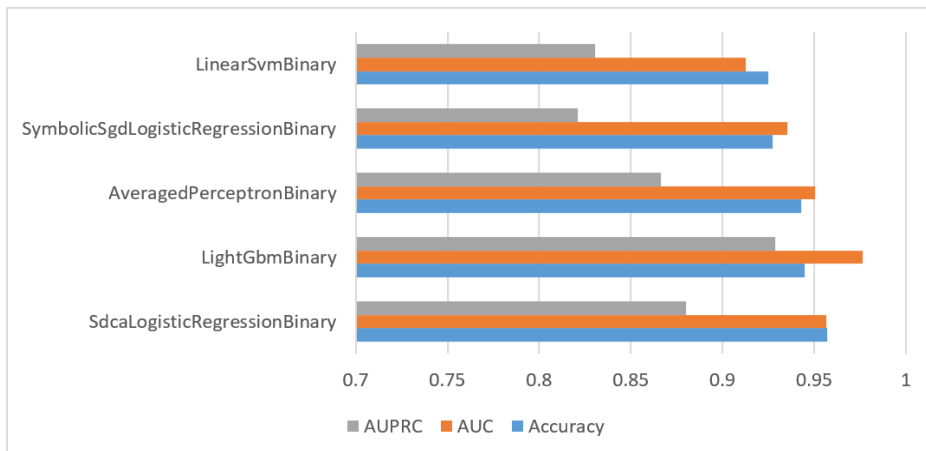
Longer autotrain sessions were also tried, but there was no better result returned from the AutoML system than SDCA Logistic Regression Binary.

The final step of the AutoML subsystem is generating a usable trained model for consumption for the .NET platform. The result is a ZIP file consisting of a model and a sample code fragment showing the example usage of the code. The code and the trained model were later used in the deployment step of the proposed solution.

Finally, the system trained on the final dataset has been tested on the smaller, basic dataset, achieving an accuracy of 94.79%.



**Fig. 4. Results for 60 seconds run**



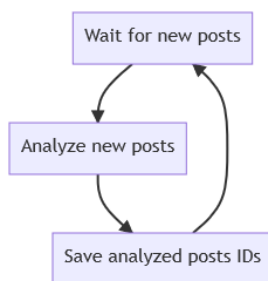
**Fig. 5. Results for 180 seconds run**

### 3. DEPLOYMENT AND RESULTS

The model has been included in a “bot” system, which is using a real user account on a system and is posting a comment to a post where the misformatted code has been found – usually because the user forgot to mark the code snippet properly, but just copying and pasting it into the text.

The first test deployment started in October 2019, and the system was limited only to report the found unmarked code problems to the logging system, no actual messages to users were sent. The problems reported were then manually checked and the algorithm was modified to post a comment only if a confidence score for a classification was greater than 0.99, to limit the number of false positives, as the author decided it would be better to not nag the user (as other, real human users can do this) than to nag the user unnecessarily.

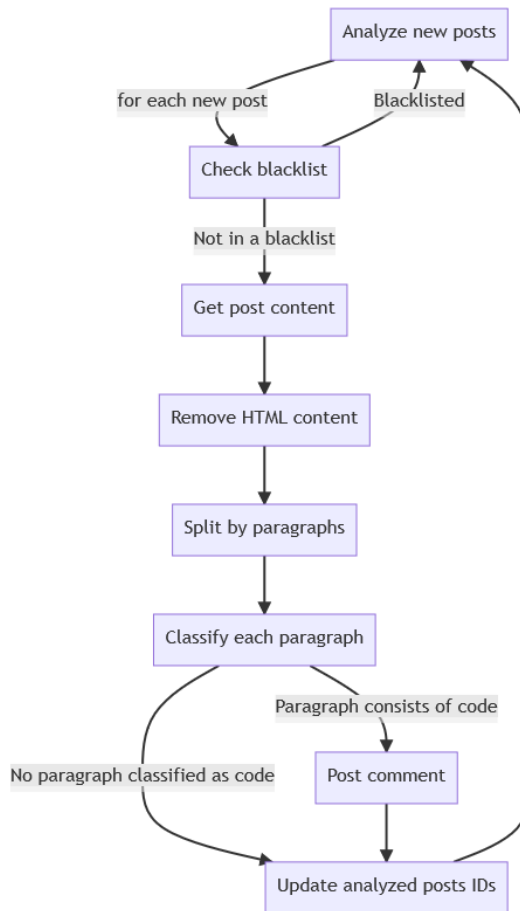
The final deployment started in early 2020. The system was prepared as a service completely separate from the discussion forum itself, running on a separate machine and interacting with API only. The overall algorithm for deployed software is presented in figure 6.



**Fig. 6. Overall system architecture**

The software is waiting for new posts for a defined time limit, which was set to 15 minutes, then it is analyzing new posts, and saves their IDs to the local datastore, so no new forum post is analyzed more than once.

For each new forum post, the software is checking if the post is on a blacklist – currently blacklisted are some of the subforums, the system should be checking only the code in “technical” subforums, discussing programming languages, not careers or educational materials. Then, the post content is being acquired from the API, and the HTML content is being removed – when the code is properly marked as a code, it is being encased in HTML5 code element, so properly marked codes will be removed in this step. Here additional checks are being run, e.g., removing some of the HTML anchor elements, which also may be classified as code, but they are not actual code snippets left by the user.



**Fig. 7. The analyzer flowchart**

Next, the content is being split into paragraphs and each paragraph is being classified by the ML system – if any paragraph of text classification is “code” with a confidence score greater than 0.99 the bot is sending a request to the forum software, creating a new comment to the post saying, “This post may include unformatted code.” and link to a help webpage describing the problem.

In the last step, no matter if the unmarked code was found or not, the post ID is saved into the “already analyzed” dataset to prevent double comments.

## 4. CONCLUSIONS

The authors tried to solve a very specific problem of differentiating between the source code of a programming language and a technical text in the case of very short snippets. Such a problem is not widely discussed, as similar papers are mostly concentrating on differentiating between different programming languages.

The proposed analyzer was built using the AutoML system included in the ML.NET library and then deployed. Until the half of 2021, the deployed solution analyzed over 100,000 posts and published over 500 comments when unmarked code was found. Users of the forum have been informed that the system is in use and were able to report the situation that the system marked the post as unformatted while there was no such case (“false positive”) – until the half of 2021, the number of false positives reports was 12, resulting in False Positive Rate of 2.4%.

**Tab. 1. Accuracy of differentiating between code and text in case of short code snippet**

Library	Accuracy
pygments	45.84%
linguist	1.01%
Proposed solution	94.79%

When comparing the proposed solution with broadly available alternatives, which are suited for a bit more advanced case, the difference in accuracy is astonishing.

To conclude, the system is working properly and living up to the expectations. The AutoML process was easy to use and generated a working model, available to use without much hassle. The author believes that AutoML systems are an interesting case in trying to apply Machine Learning to solve problems where neither the highest possible accuracy nor the knowledge of tuning and applying data models is very important. The particular problem may be exchanged and implemented into a software development process for documentation classification and archiving purposes, but the whole AutoML ecosystem is very useful and may be implemented in the industry in overall.

The software, generated models, datasets, and training logs for the deployed analyzer are available (Badurowicz, 2020) as the Free Software on the GitHub platform.

## REFERENCES

- 4programmers.net. (2000). *Forum dyskusyjne dla programistów*. <https://4programmers.net>
- Ahmed, Z., Amizadeh, S., Bilenko, M., Carr, R., Chin, W.-S., Dekel, Y., Dupre, X., Eksarevskiy, V., Filipi, S., Finley, T., Goswami, A., Hoover, M., Inglis, S., Interlandi, M., Kazmi, N., Krivosheev, G., Luferenko, P., Matantsev, I., Matushevych, S., Moradi, S., Nazirov, G., Ormont, J., Oshri, G., Pagnoni, A., Parmar, J., Roy, P., Siddiqui, M. Z., Weimer, M., Zahirazami, S., and Zhu, Y. (2019). Machine Learning at Microsoft with ML.NET. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2448–2458). Association for Computing Machinery. <https://doi.org/10.1145/3292500.3330667>
- Alreshedy, K., Dharmaretnam, D., German, D. M., Srinivasan, V., & Gulliver, T. A. (2018). *SCC: Automatic Classification of Code Snippets*. arXiv:1809.07945. <https://doi.org/10.48550/arXiv.1809.07945>
- Badurowicz, M. (2020). *ktos/Eleia: 4programmers.net bot for nagging users when their code in post is not marked as code*. <http://github.com/ktos/eleia>

- Van Dam, J. K., & Zaytsev, V. (2016). Software Language Identification with Natural Language Classifiers. *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)* (pp. 624–628). IEEE. <https://doi.org/10.1109/SANER.2016.92>
- Gilda, S. (2017). Source code classification using Neural Networks. *2017 14th International Joint Conference on Computer Science and Software Engineering (JCSSE)* (1–6). IEEE. <https://doi.org/10.1109/JCSSE.2017.8025917>
- GitHub Copilot – Your AI pair programmer. (n.d.). Retrieved January 22, 2021 from <https://copilot.github.com>
- He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212, 106622. <https://doi.org/https://doi.org/10.1016/j.knosys.2020.106622>
- Khasnabish, J. N., Sodhi, M., Deshmukh, J., & Srinivasaraghavan, G. (2014). Detecting Programming Language from Source Code Using Bayesian Learning Techniques. In P. Perner (Ed.), *Machine Learning and Data Mining in Pattern Recognition* (pp. 513–522). Springer International Publishing.
- Kłosowski, G., Kulisz, M., Lipski, J., Maj, M., & Bialek, R. (2021). The Use of Transfer Learning with Very Deep Convolutional Neural Network in Quality Management. *European Research Studies Journal*, XXIV(Special Issue 2), 253–263. <https://doi.org/10.35808/ersj/2222>
- Kulisz, M., Kujawska, J., Przysucha, B., & Cel, W. (2021). Forecasting Water Quality Index in Groundwater Using Artificial Neural Network. *Energies*, 14(18), 5875. <https://doi.org/10.3390/en14185875>
- LeClair, A., Eberhart, Z., & McMillan, C. (2018). Adapting Neural Text Classification for Improved Software Categorization. *2018 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (461–472). IEEE. <https://doi.org/10.1109/ICSME.2018.00056>
- Linguist. (n.d.). Retrieved January 22, 2022 from <https://github.com/github/linguist>
- Machrowska, A., Szabelski, J., Karpiński, R., Krakowski, P., Jonak, J., & Jonak, K. (2020). Use of Deep Learning Networks and Statistical Modeling to Predict Changes in Mechanical Parameters of Contaminated Bone Cements. *Materials*, 13(23), 5419. <https://doi.org/10.3390/ma13235419>
- Madani, N., Guerrouj, L., Di Penta, M., Gueheneuc, Y.-G., & Antoniol, G. (2010). Recognizing Words from Source Code Identifiers Using Speech Recognition Techniques. *2010 14th European Conference on Software Maintenance and Reengineering* (pp. 68–77). IEEE. <https://doi.org/10.1109/CSMR.2010.31>
- Ohashi, H., & Watanobe, Y. (2019). Convolutional Neural Network for Classification of Source Codes. *2019 IEEE 13th International Symposium on Embedded Multicore/Many-Core Systems-on-Chip (MCSoc)* (pp. 194–200). IEEE. <https://doi.org/10.1109/MCSoc.2019.00035>
- Pygments - Python syntax highlighter. (n.d.). Retrieved January 22, 2021 from <https://pygments.org>
- Sobaszek, L., Gola, A., & Kozłowski, E. (2020). Predictive Scheduling with Markov Chains and ARIMA Models. *Applied Sciences*, 10(17), 6121. <https://doi.org/10.3390/app10176121>
- Szabelski, J., Karpiński, R., & Machrowska, A. (2022). Application of an Artificial Neural Network in the Modelling of Heat Curing Effects on the Strength of Adhesive Joints at Elevated Temperature with Imprecise Adhesive Mix Ratios. *Materials*, 15(3), 721. <https://doi.org/10.3390/ma15030721>
- Ugurel, S., Krovetz, R., & Giles, C. L. (2002). What's the Code? Automatic Classification of Source Code Archives. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 632–638). ACM Digital Library. <https://doi.org/10.1145/775047.775141>
- Wever, M., Tornede, A., Mohr, F., & Hullermeier, E. (2021). AutoML for Multi-Label Classification: Overview and Empirical Evaluation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 43(09), 3037–3054. <https://doi.org/10.1109/TPAMI.2021.3051276>
- Yin, P., Deng, B., Chen, E., Vasilescu, B., & Neubig, G. (2018). Learning to Mine Aligned Code and Natural Language Pairs from Stack Overflow. *International Conference on Mining Software Repositories* (pp. 476–486). ACM Digital Library. <https://doi.org/10.1145/3196398.3196408>

*Keywords: mammography, transfer learning,  
computer vision, image processing*

*Muayed S AL-HUSEINY* [0000-0003-3192-0232]\*, *Ahmed S SAJIT* [0000-0002-1894-9033]\*\*

## **BREAST CANCER CAD SYSTEM BY USING TRANSFER LEARNING AND ENHANCED ROI**

### **Abstract**

*Computer systems are being employed in specialized professions such as medical diagnosis to alleviate some of the costs and to improve dependability and scalability. This paper implements a computer aided breast cancer diagnosis system. It utilizes the publicly available mini MIAS mammography image dataset. Images are preprocessed to clean isolate breast tissue region. Extracted regions are used to adjust and verify a pretrained convolutional deep neural network, the GoogLeNet. The implemented model shows good performance results compared to other published works with accuracy of 86.6%, sensitivity of 75% and specificity of 88.9%.*

### **1. INTRODUCTION**

Breast cancer is one of the most frequently occurring types of cancers (Batra, Sekhar & Radha, 2020). It is reported that in the year 2018 alone more than two million new cases were diagnosed (Breast Cancer Facts and Statistics, 2018). A study conducted by Cancer research UK showed that almost all women diagnosed with breast cancer at earlier stages are most likely to live for at least 5 years after their diagnosis compared to only 15% of those diagnosed at later stages (Survival, 2018). A recent report sponsored by Breast Cancer Care revealed that more than 40% of UK National Health Service (NHS) trusts lack the qualified specialist nurses to manage breast cancer (Written evidence (RTR0073), n.d.). Further, it is reported that the survival rates of cases of breast cancer in developing countries is under 40% with the main factor for such low rates being the scarcity of early diagnosis programs as well as the lack of specialized facilities and trained staff (Breast cancer: prevention and control, 2008). These findings together with the fact that this type of cancer is invasive necessitates particular attentiveness of medical and technological communities alike (Batra et al., 2020). This is an important area of research because during conventional examination by a radiologist, non-cancerous lesions can be misclassified as a cancer (false-positive), while malignancies may be missed (false-negative) resulting in radiologists failing to detect between 10% and 30% of breast cancers cases. Computer aided diagnosis (CAD) is aimed to speed up and enhance the diagnosis process via assisting specialists in the detection and classification of the breast cancer by deploying scalable computerized diagnostic tools,

---

\* Wasit University, Department of Electrical Engineering, Iraq, malhuseiny@uowasit.edu.iq

\*\* Wasit University, College of Engineering, Iraq, aalhuseiny@uowasit.edu.iq

hence, restricting the occurrence of human related shortcomings (AL-Huseiny & Sajit, 2021; Batra et al., 2020; Jalalian et al., 2013; Tan, Sim & Ting, 2017). The introduction of computerized approaches into medical procedures is highly reliant on the sensing mechanisms. Depending on the particular clinical condition and region in the human body, medical doctors utilize among a pool of diagnostic tools starting from physical inspection, patient history, lab tests and extending to sophisticated imaging procedures such as x-ray, ultrasonography, CT scans, MRI, mammography and etc. Successful exploitation of modern powerful computation in CAD is closely related to the development of computer algorithms to work with each of these diagnostic tools, particularly, different imaging modalities (Yadav & Yadav, 2020). Mammography is one of the most widely used method in the diagnosis of breast cancer. It is a relatively inexpensive screening procedure that usually takes between 10 and 15 minutes to perform (Davis, 2021). In most computer imaging procedures, CAD systems are constituted of four parts: (a) preprocessing, (b) detection of regions of interest (ROI), (c) feature selection/extraction, and (d) classification (Jalalian et al., 2013). The reliability of conventional classification methods resides in the implementation of a suitable data representation. Much of the efforts in this regard are dedicated to the identification of discriminative traits, a difficult and time-demanding operation that requires the knowledge of domain experts (Spanhol et al., 2016a). Many CAD approaches employ a combination of hand-designed heuristic features with mathematical descriptor features (Zeiler, Taylor & Fergus, 2011). An alternative path is to derive features automatically from training images (Lazebnik, Schmid & Ponce, 2006). Recent advancements allowed machine learning/computer vision tasks to perform feature learning feasibly. Convolutional neural networks (CNN) which perform local connectivity pattern among neurons of successive layers can characterize spatial locality of input features yielding highly dependable feature extraction. Combined with the inherent powerful generalization capability of deep neural networks, these together are projected to form a strong CAD system candidate for breast cancer (Batra et al., 2020).

## 1.1. Literature Review

In previous related studies, the authors in (Batra et al., 2020) used preprocessing to enhance input images and followed that by image segmentation to extract strips of the malignant tissue. These are then fused with the original images to train their 8 layers custom-built CNN. They also studied the trade-off between accuracy and training in various execution environment such as Tensorflow and Matlab. The authors in (Jamieson, Drukker & Giger, 2012) modified the Adaptive Deconvolutional Networks (Zeiler, Taylor & Fergus, 2011) into multiple layers of representation to learn two datasets of breast image data: full field digital mammography (FFDM) and ultrasound sets. They employed Spatial Pyramid Matching to the inferred feature maps and used linear support vector machine for classification to achieve moderate performance. CNNs are also used in detection and analysis using histopathology images, the authors in (Zainudin, Shamsuddin & Hasan, 2021) proposed four types of deep layer CNN architecture which are called 6-layer CNN, 13-layer CNN, 17-layer CNN and 19-layer CNN, respectively, to detect malignant cells (mitosis) in breast tissue by using histopathology image and to compare the performance of different layered implementations. Spanhol et al (Spanhol et al., 2016a) also implemented a deep learning approach to classify breast cancer histopathological images taken from the publicly



available BreKHis dataset (Spanhol et al., 2016b). They suggested a method to extract image patches to train the CNN and then combined these patches with the original images for final classification. The study employs transfer learning and also combines different CNNs by using simple fusion rules with some reported improvement accuracy. Masud et al (Masud, Eldin Rashed & Hossain, 2020) in their research also consider transfer learning. They fine-tuned eight different pre-trained models as well as a proposed shallow convolutional neural network to compare the performance of these models in classifying breast cancers on two merged ultrasound datasets. Noise was removed by applying morphological operations to extract the ROIs. The achieved results show that the performance of the proposed CNN outperforms that of the pre trained models. Arevalo et al. (Arevalo et al., 2016) suggested a new representation learning framework for the diagnosis of breast cancer in mammograms which merges deep learning with the automatic learning of features. They built a new biopsy proven dataset and followed a hybrid approach, hence, CNNs are used to learn the features in a supervised setting. Petersen et al (Petersen et al., 2014) implemented a convolutional sparse autoencoder (CSAE) method which extracts descriptive features from unlabeled mammograms taken from the Dutch biennial breast screening program dataset (Otten et al., 2005). The learned features are used to train a simple classifier which performs tissue segmentation, scoring of percentage mammographic density (PMD), and scoring of mammographic texture (MT). The authors in (Tan, Sim & Ting, 2017) employed Google's TensorFlow to develop their classifier on mini-MIAS mammogram dataset. They used preprocessing to crop abnormality tissues to 48 x 48 and then fed those features to CNN classifier as training source. Authors in (Charan, Khan & Khurshid, 2018) suggested a transfer learning model based on GoogleNet and AlexNet pre-trained models, they also utilized preprocessing techniques. Their proposed model is applied to mammograms with segmented cancer regions. The authors claim that the model achieves improved performance over human involved methods.

The research problem addressed in this paper is to design a learning based computer aided breast cancer diagnosis system. This is pursued building on previous efforts by implementing CAD system where preprocessing is used to rid input images of markings, edges, and unrelated body tissues. ROI containing only breast region is then segmented and used to adjust and test a pretrained deep CNN. the rest of this paper includes, in section two the material and methods used, the results in section three and finally the conclusions in section four.

## **2. MATERIALS AND METHODOLOGY**

The proposed framework consists of a set of pre-processing steps, followed by the setup of a machine learning model. The model is trained and validated by using mammographic image dataset. The various stages of the framework are presented subsequently.

### **2.1. MIAS Dataset**

This digital dataset was generated by the (Mammographic Image Analysis Society-MIAS (Suckling et al., 1994) which is a research body focused on the development and understanding of mammograms. Images produced via the (UK National Breast Screening Program) were used for the generation of the dataset after digitization of the original scans.

The digital images are utilized in a 50-micron pixel edge by using a Joyce-Loebl microdensitometer at 8-bit resolution per pixel (see Fig. 1-a and Fig. 2-a). The dataset includes a total of 322 digital images (161 patients) along with expert radiologist annotations specifying the regions of detected abnormalities. The dataset is publicly accessible through the Pilot European Image Processing Archive (PEIPA) at the University of Essex (Grgic et al., 2021). The provided information describing the dataset includes:

1. The type of the background tissue: fatty, fatty-glandular and dense-glandular.
2. The detected abnormality: calcification; circumscribed masses, etc.
3. The degree of abnormality: benign and malignant.
4. The coordinates of the abnormality on the digitized image.
5. The radii of the circles surrounding the lesions.

The image files in the dataset are organized in consecutive odd/even numbered pairs representing respectively the right and left mammograms of individual patients. The images were normalized to the center of the 1024x1024 grid with origin represented by the lower left element of the grid.

## 2.2. Pre-processing

In most classification applications, preprocessing plays a crucial role in exposing and magnifying discriminatory features. It is used in this research to perform two functions: to rid unnecessary markings and human tissues, and to normalize the appearance of all images. As such the resulting images will be a relatively clean centered region of interest (ROI) with normalized size and intensity for all files of the used dataset. The stages of preprocessing are summarized:

Gabor filter (Aach, Kaup & Mester, 1995) is used to characterize differently textured tissue parts. This will help remove unrelated markings and structures. Gabor filters are bandpass filters frequently used in image processing applications, particularly, in dealing with human tissues as texture is a more discriminant trait than other features such as shape, brightness or etc. Eq. (1) gives the response of Gabor mask to an input image  $X(r, c)$ :

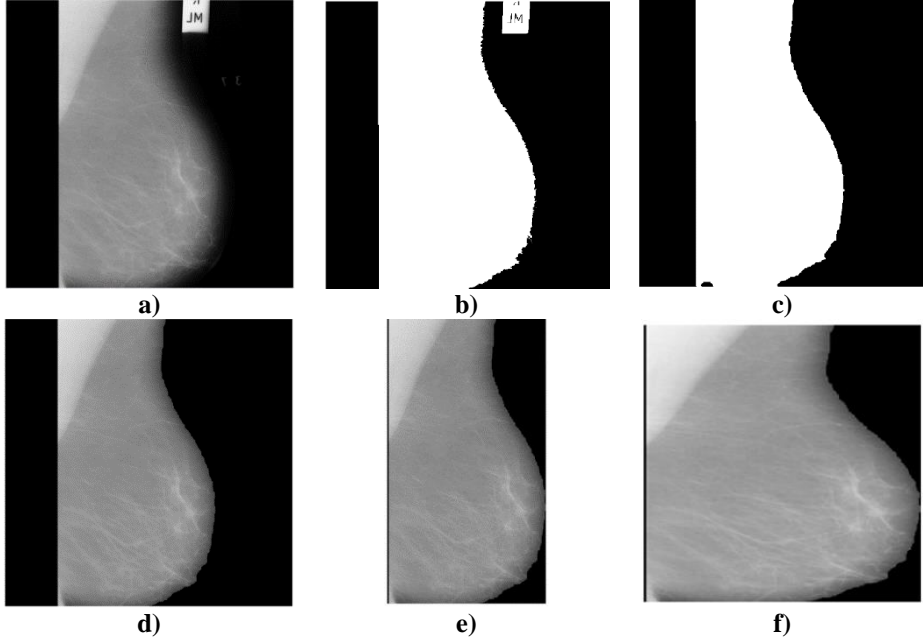
$$G_{rc} = \exp\left(-\frac{\hat{r}^2 + \gamma^2 \hat{c}^2}{2\sigma^2}\right) \cos\left(2\pi \frac{\hat{r}}{\lambda} + \psi\right) \quad (1)$$

where  $\hat{r} = r \cos \theta + c \sin \theta$ , and,  $\hat{c} = r \sin \theta + c \cos \theta$  of  $X(r, c)$ . Filter parameters are:  $\lambda$  represents the wavelength,  $\theta$  is the angle of the normal to the wave of a Gabor function,  $\psi$  is the phase shift,  $\sigma$  is the Gaussian standard deviation and  $\gamma$  is the aspect ratio (AL-Huseiny & Sajit, 2021; Gabor filter, n.d.). The values of these parameters are tuned to achieve best results with respect to the input images in terms of emphasizing unwanted structures to be eliminated by later stages of preprocessing. the output of Gabor filter is thresholded as shown in Fig. 1-b and Fig. 2-b.

Morphological operations: after Gabor filter, the images are treated with morphological operators with carefully selected structuring element. This helps remove small and densely perforated regions after texture analysis, including markings and muscle tissue as depicted in the examples of Fig. 1-c and Fig. 2-c. morphological operation is given in Eq. (2):

$$G_{Mrph} = Morph_{sel}(G) \quad (2)$$

where  $G_{Mrph}$  is the output of morphological operator  $Morph$  performing under the structuring element  $sel$ .



**Fig. 1. Stages of preprocessing – example 1: a) input image, b) texture analysis followed by thresholding, c) morphological operation and region selection, d) masking of original image, e) cropping to ROI, and f) size and intensity normalization**

Region selection: in cases where more than one region is left after morphological operations, the region with the largest area is preserved while other regions are omitted from the binary image forming a mask  $G_{Msk}$  for the desired ROI.

Masking: input images are multiplied by their respective masks to extract their ROIs as given by Eq. (3). This is shown in Fig. 1-d and Fig. 2-d.

$$X_{ROI} = \sum_r X(r, c) * G_{Msk}(r, c) \quad (3)$$

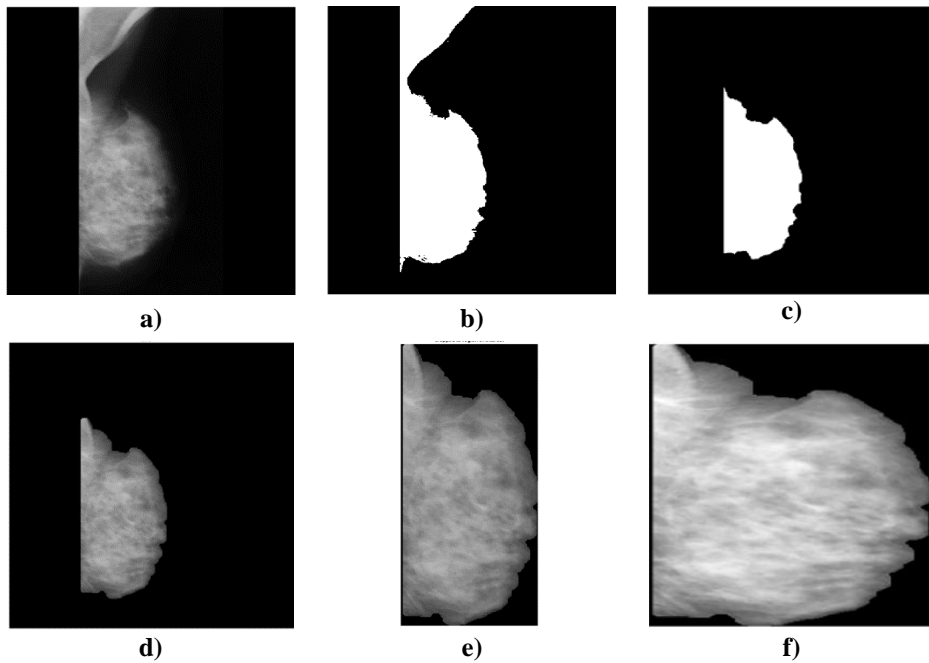
Cropping: masked images are then cropped to the boundaries of the extracted region to give  $X_{Crpd}$ . The results of this stage can be seen in Fig. 1-e and Fig. 2-e

Normalization: this step includes both size and intensity normalization, thus, each cropped image of ROI is scaled to 224X224 pixels image  $X_{Scl}$ . Scaled images are then intensity normalized according to Eq. (4) (Gonzalez & Woods, 2006):

$$X_{Nrm} = (X_{Scl} - Min) \frac{newMax - newMin}{Max - Min} + newMin \quad (4)$$

where  $X_{Scl}$  and  $X_{Nrm}$  are scaled and output normalized images respectively,  $Min$  and  $Max$  are minimum and maximum values of the input scaled image,  $newMin$  and  $newMax$  are

the desired minimum and maximum values of the normalized image, in this research these values are set to 0 and 1 respectively. The output of this stage is presented in the examples of Fig. 1-f and Fig. 2-f.

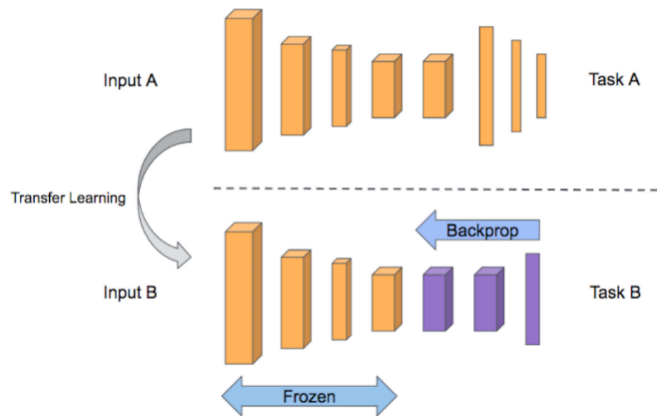


**Fig. 2. Stages of preprocessing – example 2: a) input image, b) texture analysis followed by thresholding, c) morphological operation and region selection, d) masking of original image, e) cropping to ROI, and f) size and intensity normalization**

### 2.3. Transfer learning with GoogLeNet

Transfer learning involves the use of previously developed computer models to learn patterns of new data different from that used in the development of these models (AL-Huseiny & Sajit, 2021). This approach is useful particularly when the available resources such as data and computing power are limited. The attractiveness of such strategy should not distract from the fact that not all trained models are capable of learning previously unseen data. Thus, the process involves some sort of tuning to parts of the transferred models to ensure that there is a compatibility in terms of data types as well as expected classes. In Fig. 3 it is noticed that early layers are largely kept unchanged due to the fact that these layers are activated on low level features and this is the type of knowledge desirable to be transferred to the new domain. Subsequent and last layers are obviously replaced and trained with the new data (domain), the reason is that these layers model data specific attributes, as such, they are activated by high level features (AL-Huseiny, Abbas & Sajit, 2020; AL-Huseiny & Sajit, 2021; Al-Yasriy et al., 2020). The use of transfer learning strategy significantly diminishes training time as large parts of the network are at or near convergence.

Convolutional neural networks (CNNs) are a category of deep NNs widely utilized in the analysis of images and videos. These models are characterized by being shift/space invariant in the sense that their convolution filters which slide along input features use shared-weight architecture. This configuration gives translation equivariant responses commonly defined as feature maps ( Convolutional neural network, n.d.; Zhang, 1990).



**Fig. 3. The concept of transfer learning shown in terms of model layers – only learnable layers are readjusted (Malgonde, 2021)**

GoogLeNet is a Convolutional NN developed by Szegedy et al (Szegedy et al., 2015) for the ImageNet challenge (Russakovsky et al., 2015). This challenge involves the development of high feature representations by millions of images of ordinary objects collected in a huge dataset. The network architecture is formed mainly of stacking of Inception modules as shown in Fig 4. The layers of this deep CNN are listed in Table 1. As a standard approach in computer vision with deep learning, this CNN classifies input images into one of the learned classes and produces a value representing the level of confidence of the output. The configuration of this CNN ensures that it has 12 times less parameters than Alexnet (Santos, 2019). The structure of GoogLeNet constitutes 22 layers. Notably, 9 prominently important constructions called the inception modules.

Inception modules depicted in Fig. 5 employ learnable filters ranging in size from  $(1 \times 1)$  to  $(5 \times 5)$  and are specialized in emphasizing various level features simultaneously (Szegedy et al., 2015). These structures act on analyzing the correlation statistics coming out of the activations of preceding layers. Correlated activations are clustered together in one vector (Tripathy, 2016). This configuration is well suited for computer vision applications where higher correlations occur in local neighborhoods, hence, the use of  $1 \times 1$  filters. As for general features spread across larger neighborhoods, larger filters will capture their correlations.



## 2.4. Experimental

The current study employs the digitized images of the mini MIAS mammographic slides (Suckling et al., 1994). Depending on the specialists' diagnosis, this sample divides its cases into three categories: Normal (NRM), Benign (BGN) and Malignant (MLG). Images are initially prepared as detailed in Sec 2.2 to extract the ROI. The sample is then grouped into two sets for training and testing at the ratio of 70:30. The larger set is used to train the transferred GoogLeNet model and adjust its learnable weights. The second set is used to test the performance of the learned model. In this setting, the loss3 classifier (fully connected layer) and the output (classification) layers are replaced and fine-tuned by using the training data to accommodate data specific features and to produce the desired number of classes respectively.

<b>Results</b>	
Validation accuracy:	86.60%
Training finished:	Reached final iteration
<b>Training Time</b>	
Start time:	27-May-2021 06:55:57
Elapsed time:	56 min 20 sec
<b>Training Cycle</b>	
Epoch:	30 of 30
Iteration:	2160 of 2160
Iterations per epoch:	72
Maximum iterations:	2160
<b>Validation</b>	
Frequency:	72 iterations
<b>Other Information</b>	
Hardware resource:	Single CPU
Learning rate schedule:	Constant
Learning rate:	3e-05

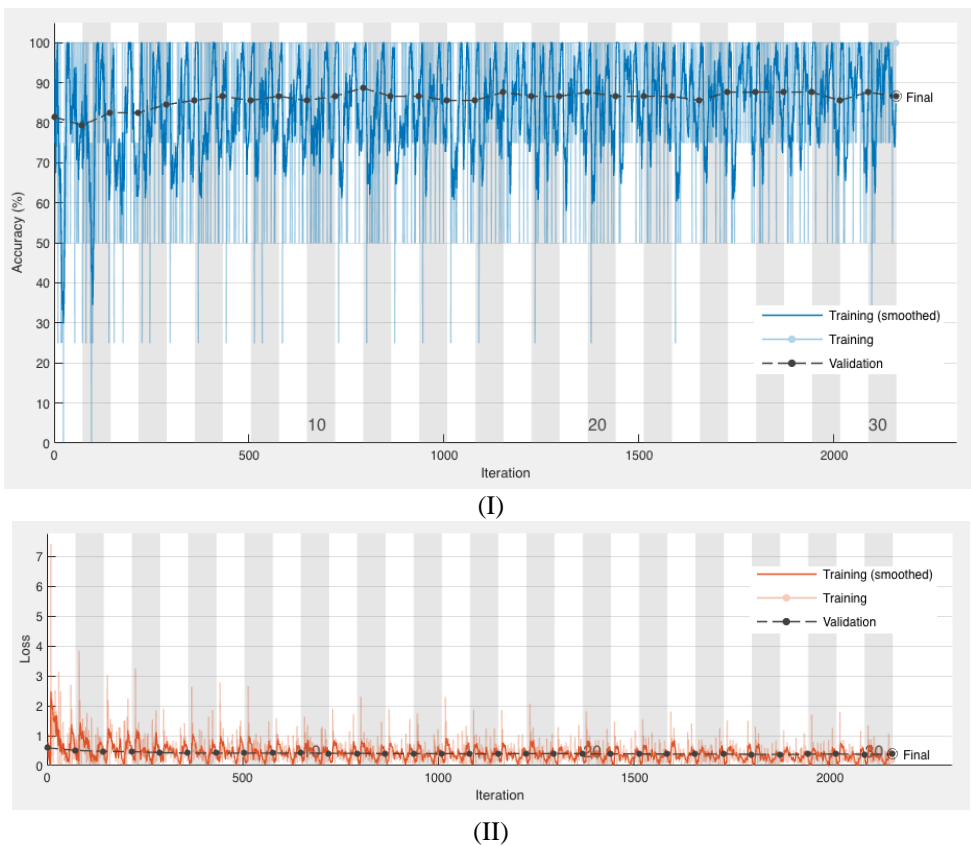
**Fig. 6. MATLAB user interface showing the parameters used in the training/testing of the deep neural network**

For learning the model, the number of epochs was set to 30 with stochastic gradient descent (SGD) learning strategy. The learning rate was set to  $3 \times 10^{-5}$  except for the last layers which was at  $3 \times 10^{-4}$ , these values were chosen empirically. In order to assure better learning images were shuffled at the start of the epoch. Further, and avoid overfitting these images were augmented by using random scaling, translation and rotation. The model was implemented by using MATLAB® R2020b deep learning toolbox on a dual-core Intel Core i5 MacBook pro machine clocked at 2.5 GHz with 16 GB DDR3 RAM. The algorithm parameters were adjusted to the values listed in Fig. 6.

### 3. DATA ANALYSIS AND INTERPRETATION

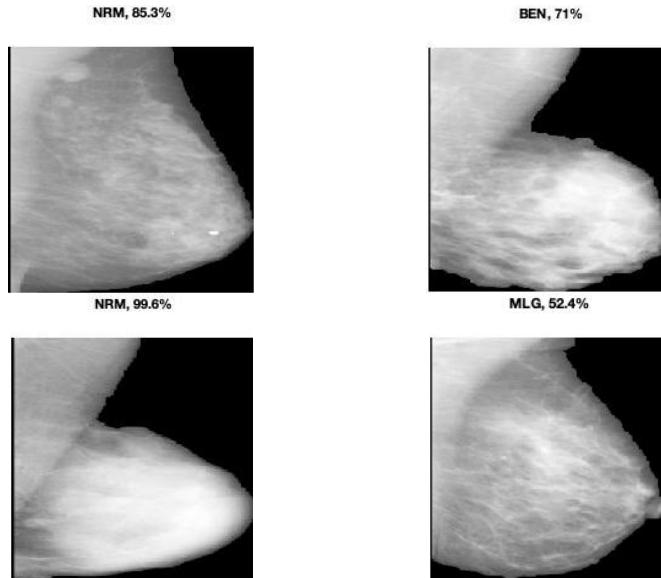
The transfer learning approach employed in this research to use a pretrained deep neural network, the GoogLeNet was used to classify the mammograph images of the mini MIAS breast cancer dataset. The results show that the transfer learned model has achieved an accuracy of 86.6% for the test sample. In Fig. 7 the accuracy and the loss over the course of training and testing over 2160 iterations is shown. It is obvious that the algorithm begins settling after 1100 iterations. In the graphs of Fig. 7 continuous ripple is observed in the accuracy as well as the loss, this can be referred to the presence of noise and non-uniform imaging conditions common in real world data as well as the augmentation deployed to overcome overfitting.

A group of four randomly selected classified images is shown in Fig 8. The figure also provides the probability of the generated class for each image. It is evident that the algorithm in some cases is making guesses as would be the case with specialists. However, the majority of classifications is performed with high degree of confidence. The trained model also scores sensitivity of 75% and specificity of 88.9%, further detailed outcomes are presented in the confusion matrix of Tab. 1.



**Fig. 7. The outcomes of the training/testing of the proposed transfer learning model over 30 epochs and 2160 iterations in terms of (I) accuracy, (II) loss**





**Fig 8: Examples of classified mammograph images selected at random, shown with the classification labels are the probabilities of the generated classes**

The outcomes of the proposed implementation are measured against other related models mentioned in the literature. Tab. 3 provides comparison figures for the proposed system against those models, in which it shows that this work scored higher accuracy, and also, better specificity (where values were provided), which indicates an improved true negative and false positive rates. Sensitivity was slightly below that of other methods which refers to higher false negative cases produced by this approach compared to the other reported methods. Despite being better on other metrics, this aspect needs to be looked into and improved perhaps by using larger datasets or merging multiple datasets together to help the model learn deeper features in order to be more discriminative.

**Tab. 2. Confusion Matrix**

Confusion matrix		Predicted class	
		Non-malignant (positive)	Malignant (negative)
Actual class	Non-malignant	TP = 12	FN = 4
	Malignant	FP = 9	TN = 72

**Tab. 3. Comparison of methods**

Method	Dataset	Accuracy	Sensitivity	Specificity	Time
Batra (Batra, Sekhar & Radha, 2020)	mini MIAS	84.02%	/	/	45 minutes
BCDCNN (Tan, Sim & Ting, 2017)	mini MIAS	71%	82.68%	82.73%	/
Spanhol (Spanhol et al., 2016a)	BreaKHis	84%	/	/	/
Charan (Charan, Khan & Khurshid, 2018)	MIAS	65%	/	/	/
Zainudin (Zainudin, Shamsuddin & Hasan, 2021)	MITOS-ATYPHIA	84.49 %	80.55 %	/	/
<b>Proposed</b>	mini MIAS	86.6%	75%	88.9%	56 minutes

## 4. CONCLUSION

In this paper a computer aided diagnosis system for breast cancer is presented which employs a learning algorithm to model the data of publicly published mammographic images, the mini MIAS dataset. The images were initially preprocessed to clean and extract breast tissue only ROI and remove markings and clutter. They are then size and brightness normalized. The treated dataset was then augmented to compensate for the limited size of the sample. The set is then fed to the GoogLeNet CNN in a transfer learning approach to save training time and reuse derived features. Initial and final layers of the CNN were replaced to accommodate input data and the desired classes. This setting produced an accuracy of 86.6% which is better than those of the methods reported previously. Furthermore, this approach is easy to setup and modify. Therefore, compared to human experts it is scalable, reportedly more stable and less prone to subjective and circumstantial influences.

## REFERENCES

- Aach, T., Kaup, A., & Mester, R. (1995). On texture analysis: Local energy transforms versus quadrature filters. *Signal Processing*, 45(2), 173-181. [https://doi.org/10.1016/0165-1684\(95\)00049-J](https://doi.org/10.1016/0165-1684(95)00049-J)
- AL-Huseiny, M. S., Abbas, N. K., & Sajit, A. S. (2020). Diagnosis of arrhythmia based on ECG analysis using CNN. *Bulletin of Electrical Engineering and Informatics*, 9(3), 988-995. <https://doi.org/10.11591/eei.v9i3.2172>
- AL-Huseiny, M. S., & Sajit, A. S. (2021). Transfer learning with GoogLeNet for detection of lung cancer. *Indonesian Journal of Electrical Engineering and Computer Science*, 22(2), 1078-1086. <http://doi.org/10.11591/ijeecs.v22.i2.pp1078-1086>
- Al-Yasriy, H. F., Al-Husieny, M. S., Mohsen, F. Y., Khalil, E. A., & Hassan, Z. S. (2020). Diagnosis of lung cancer based on CT scans using CNN. *IOP Conference Series: Materials Science and Engineering*, 928, 022035. <https://doi.org/10.1088/1757-899x/928/2/022035>
- Arevalo, J., González, F. A., Ramos-Pollán, R., Oliveira, J. L., & Guevara Lopez, M. A. (2016). Representation learning for mammography mass lesion classification with convolutional neural networks. *Computer Methods and Programs in Biomedicine*, 127, 248-257. <https://doi.org/10.1016/j.cmpb.2015.12.014>
- Batra, K., Sekhar, S., & Radha, R. (2020). Breast cancer detection using CNN on mammogram images. *Computational Vision and Bio-Inspired Computing* (pp. vol 1108). Springer. [https://doi.org/10.1007/978-3-030-37218-7\\_80](https://doi.org/10.1007/978-3-030-37218-7_80)
- Breast Cancer Facts and Statistics. (2018). Retrieved June 12, 2021 from <https://www.breastcancer.org/facts-statistics>
- Breast cancer: prevention and control. (2008). World Health Organisation. <https://www.who.int/cancer/detection/breastcancer/en/index1.html#:~:text=Breast%20cancer%0survival%20rates%20vary,et%20al.%2C%202008>
- Charan, S., Khan, M. J., & Khurshid, K. (2018). Breast cancer detection in mammograms using convolutional neural network. *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-5). IEEE. <https://doi.org/10.1109/ICOMET.2018.8346384>
- Convolutional neural network. (n.d.). Wikipedia Retrieved June 21, 2022 from [https://en.wikipedia.org/w/index.php?title=Convolutional\\_neural\\_network&oldid=1029918158](https://en.wikipedia.org/w/index.php?title=Convolutional_neural_network&oldid=1029918158)
- Davis, L. E. (n.d.). *What Is a Mammogram?* Retrieved June 20, 2021 from <https://www.verywellhealth.com/mammogram-what-to-expect-430283>
- Deep Learning Network Part Three: GoogLeNet Series*. (n.d.). Retrieved June 15, 2021 from <https://www.programmersought.com/article/85103454206/>
- Gabor filter. (n.d.). Wikipedia. Retrieved June 21, 2022 from [https://en.wikipedia.org/w/index.php?title=Gabor\\_filter&oldid=993157632](https://en.wikipedia.org/w/index.php?title=Gabor_filter&oldid=993157632)
- Gonzalez, R. C., & Woods, R. E. (2006). *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc.
- Grgic, M., Delac, K., Bozek, J., & Rangayyan, R. M. (2021). *Mammographic image analysis homepage*. Video Communications Laboratory (VCL), Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia.

- Jalalian, A., Mashohor, S. B., Mahmud, H. R., Saripan, M. I., Ramli, A. R., & Karasfi, B. (2013). Computer-aided detection/diagnosis of breast cancer in mammography and ultrasound: a review. *Clinical Imaging*, 37(3), 420-426. <https://doi.org/10.1016/j.clinimag.2012.09.024>
- Jamieson, A. R., Drukker, K., & Giger, M. L. (2012). Breast image feature learning with adaptive deconvolutional networks. *Proc. SPIE 8315, Medical Imaging 2012: Computer-Aided Diagnosis* (no. 831506). <https://doi.org/10.1117/12.910710>
- Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond Bags of features: spatial pyramid matching for recognizing natural scene categories. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* (pp. 2169-2178). IEEE. <https://doi.org/10.1109/CVPR.2006.68>
- Malgonde, S. (2021). *Transfer learning using Tensorflow*. <https://medium.com/@subodh.malgonde/transfer-learning-using-tensorflow-52a4f6bcde3e>
- Masud, M., Eldin Rashed, A. E., & Hossain, M. S. (2020). Convolutional neural network-based models for diagnosis of breast cancer. *Neural Computing and Applications*. Springer. <https://doi.org/10.1007/s00521-020-05394-5>
- Melli, G. (2021). *GoogLeNet*. <https://www.gabormelli.com/RKB/GoogLeNet>
- Otten, J. D. M., Karssemeijer, N., Hendriks, J. H. C. L., Groenewoud, J. H., Fracheboud, J., Verbeek, A. L. M., de Koning, H. J., & Holland, R. (2005). Effect of recall rate on earlier screen detection of breast cancers based on the dutch performance indicators. *JNCI: Journal of the National Cancer Institute*, 97(10), 748-754. <https://doi.org/https://10.1093/jnci/dji131>
- Petersen, K., Nielsen, M., Diao, P., Karssemeijer, N., & Lillholm, M. (2014). Breast tissue segmentation and mammographic risk scoring using deep learning. *Breast Imaging. IWDMM 2014, Lecture Notes in Computer Science* (vol 8539). Springer. [https://doi.org/10.1007/978-3-319-07887-8\\_13](https://doi.org/10.1007/978-3-319-07887-8_13)
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211-252. <https://doi.org/10.1007/s11263-015-0816-y>
- Santos, L. (2019). *Artificial Intelligence*. GitBook.
- Spanhol, F. A., Oliveira, L. S., Petitjean, C., & Heutte, L. (2016a). Breast cancer histopathological image classification using Convolutional Neural Networks. *2016 International Joint Conference on Neural Networks (IJCNN)* (pp. 2560-2567). IEEE. <https://doi.org/10.0.4.85/IJCNN.2016.7727519>
- Spanhol, F. A., Oliveira, L. S., Petitjean, C., & Heutte, L. (2016b). A Dataset for Breast Cancer Histopathological Image Classification. *IEEE Transactions on Biomedical Engineering (TBME)*, 63(7), 1455-1462.
- Suckling, J., Astley, S., Betal, D., Cerneaz, N., Dance, D. R., Kok, S.-L., Parker, J., Ricketts, I., Savage, J., Stamatakis, E., & Taylor, P. (1994). *The Mammographic Image Analysis Society Digital Mammogram Database Excerpta Medica*. International Congress Series.
- Survival. (n.d.). Retrieved August 20, 2022 from <https://www.cancerresearchuk.org/about-cancer/breast-cancer/survival>
- Szegedy, C., Wei, L., Yangqing, J., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1-9). IEEE. <https://doi.org/10.1109/CVPR.2015.7298594>
- Tan, Y. J., Sim, K. S., & Ting, F. F. (2017). Breast cancer detection using convolutional neural networks for mammogram imaging system. *2017 International Conference on Robotics, Automation and Sciences (ICORAS)* (pp. 1-5). IEEE. <https://doi.org/10.1109/ICORAS.2017.8308076>
- Tripathy, A. (2016). *GoogLeNet Insights* slideshare.net. [https://www.youtube.com/watch?v=\\_XF7N6rp9Jw](https://www.youtube.com/watch?v=_XF7N6rp9Jw)
- Written evidence (RTR0073). (2022). Breast Cancer Now and UK Parliament. <https://committees.parliament.uk/writtenevidence/42740/pdf/>
- Yadav, S.-P., & Yadav, S. (2020). Fusion of medical images in wavelet domain: a hybrid implementation. *Computer Modeling in Engineering & Sciences*, 122(1), 303-321. <https://doi.org/10.32604/cmes.2020.08459>
- Zainudin, Z., Shamsuddin, S. M., & Hasan, S. (2021). Deep layer convolutional neural network (CNN) Architecture for breast cancer classification using histopathological images. In A. E. Hassanien (Ed.), *Machine Learning and Big Data Analytics Paradigms: Analysis, Applications and Challenges* (pp. 347-364). Springer Nature Switzerland. [https://doi.org/10.1007/978-3-030-59338-4\\_18](https://doi.org/10.1007/978-3-030-59338-4_18)
- Zeiler, M. D., Taylor, G. W., & Fergus, R. (2011). Adaptive deconvolutional networks for mid and high level feature learning. *2011 International Conference on Computer Vision* (pp. 2018-2025). IEEE. <https://doi.org/10.1109/ICCV.2011.6126474>
- Zhang, W. (1990). Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied Optics*, 29(32), 4790-4796. <https://doi.org/10.1364/AO.29.004790>