A COMBINED FINITE ELEMENT AND MACHINE LEARNING APPROACH FOR PREDICTING SPECIFIC CUTTING FORCES AND MAXIMUM TOOL TEMPERATURES DURING ORTHOGONAL MACHINING

by

Sai Manish Reddy Mekarthy

A thesis submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Master of Science in
Mechanical Engineering

Charlotte

2019

Approved by:

_____
Dr. Harish P. Cherukuri

_____
Dr. Tony L. Schmitz

_____
Dr. Konstantinos Falaggis

ABSTRACT

SAI MANISH REDDY MEKARTHY. A combined finite element and machine learning approach for predicting specific cutting forces and maximum tool temperatures during orthogonal machining. (Under the direction of DR. HARISH P. CHERUKURI)

In machining, specific cutting forces and temperature fields in the shear zones are of primary interest. These quantities depend on many machining parameters, such as the cutting speed, rake angle, tool-tip radius, uncut chip thickness, etc. The finite element method (FEM) is the tool of choice for understanding the effect, that these parameters have on the cutting forces and temperatures. However, the simulations, even in the context of a two-dimensional orthogonal machining model, are time-consuming and thus, it is difficult to generate sufficient data that covers the entire parametric space of practical interest.

The purpose of this work is to present, as a proof-of-concept, a hybrid methodology that combines finite element method and machine learning to predict specific cutting forces and maximum tool temperatures for a given set of machining conditions. The finite element method (FE method) was used to generate the training and test data, which consisted of various machining parameters and the corresponding specific cutting forces and maximum tool temperatures. The data was then used to build a neural network model that can be used for predictive purposes.

The FE models consist of an orthogonal plane-strain machining model with the workpiece being made of the aluminum alloy, Al2024-T351. The finite element package ABAQUS/EXPLICIT was used for the simulations. The chip formation was simulated by using a recent fracture-based methodology introduced by Patel and Cherukuri. Specific cutting forces and maximum tool temperatures were calculated for several different combinations of uncut chip thickness, cutting speed and the rake angle.

For the machine learning-based predictive models, artificial neural networks were selected. The neural network modeling was performed using Python with Adam as the training algorithm. Both shallow neural networks (SNN) and deep neural networks (DNN) were built and tested with various activation functions (ReLU, ELU, Tanh, Sigmoid, Linear) to predict specific cutting forces and maximum tool temperatures. The optimal neural network architecture along with the activation function that produced the least error in prediction was identified.

DEDICATION

I would like to dedicate this work to all my teachers and family members.

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Harish P. Cherukuri for his encouragement and guidance throughout my research. His passion for teaching and research motivated me in many ways. I also want to thank Dr. Tony L. Schmitz and Dr. Konstantinos Falaggis for being part of my thesis committee. I would like to thank the UNC ROI for support.

I am grateful for the constant support my friends Tejaswini and Dilip gave me throughout my master's program. I would also like to extend my gratitude to my colleagues Jaimeen, Dhanooj, Parth, and Nishant.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FEM	Finite element method
CPU	Central processing unit
DNN	Deep neural networks
ELU	Exponential linear unit
GD	Gradient descent
GPU	Graphics processing unit
ReLU	Rectified linear unit
SCG	Scaled conjugate gradient
SNN	Shallow neural networks
MTT	Maximum tool temperature
SCF	Specific cutting force
FEA	Finite element analysis
ML	Machine learning
ANN	Artificial neural networks
ALE	Arbitrary Lagrangian formulation

LIST OF SYMBOLS

α	Rake angle
$\bar{\epsilon}_d$	Equivalent plastic strain at onset of damage
$\bar{\sigma}_y$	Yield stress after the onset of damage
$\dot{\bar{u}}$	Rate of equivalent plastic displacement
σ_n	Normal stress
τ	Shear stress
a_c	Clearance angle
d	Width of the workpiece
f	Uncut chip thickness
F_c	Cutting force
G_f	Critical energy release rate
K_s	Specific cutting force
n_r	Tool nose radius
V_c	Cutting speed

CHAPTER 1: INTRODUCTION

1.1 Orthogonal machining

Orthogonal machining, shown in the Figure 1.1, is a metal cutting process in which the cutting edge of the tool in perpendicular to the work piece.

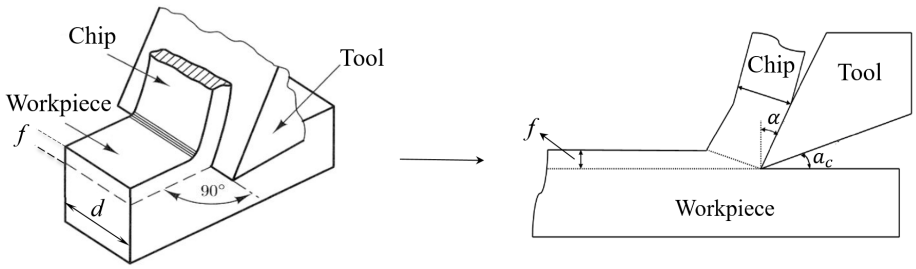


Figure 1.1: Orthogonal machining [2].

Orthogonal machining can be modeled as a 2D plane strain problem since the normal and shear strains in the lateral direction can be assumed to be zero. This is because the uncut chip thickness, f, is very small compared to the width of the workpiece d, as shown in the Figure 1.1. Because of this 2D nature it is widely used in both theoretical and experimental work, as it eliminates many independent variables.

1.1.1 Specific cutting force

The estimation of cutting forces prior to the actual machining process is important mainly for calculating the power and torque requirements. This may be best done in terms of specific cutting force (K_s) which is defined as the cutting force required to remove unit area of work material, represented by the Equation 1.1 , where F_c stands for cutting force.

$$K_s = \frac{F_c}{fd} \tag{1.1}$$

The specific cutting force will be essentially independent of cutting speed (V_c) over a wide range of values, provided a large BUE (built up edge) is not obtained [1].

The Table 1.1 presents the range of specific cutting forces for few selected work materials [5].

Table 1.1: Specific cutting forces for a range of engineering materials [1].

Material	K_c (Gpa)
Aluminum alloys	0.5-1.0
Copper alloys	1.0-2.0
Cast irons	1.5-3.0
Carbon steels	2.0-3.0
Alloy steels	2.0-5.0

1.1.2 Maximum tool temperature

Another important parameter of interest is maximum tool temperature. High temperatures are inimical to both workpiece and cutting tool as they lead to dimensional inexactitude of the workpiece due to thermal distortion, and can also cause rapid tool wear. Therefore, it is necessary to predict maximum tool temperature to improve the machining and tool life.

1.2 Machine learning

Machine learning (ML) is referred to as a process in which computers are made capable to learn repeatedly from the data sets and make proper predictions. It is an application of artificial intelligence that effectively automates the process of building models and motivates them to attune fresh scenarios autonomously. Figure 1.2 shows the classification of machine learning.

There are 2 types of ML algorithms, supervised (machines are trained by providing input-output pairs) and unsupervised (machines are provided only with input pairs and will draw their own conclusions). Supervised learning algorithms are further divided into classification and regression. In classification problems the output data sets are in the form of categories, whereas in regression problems the output data sets

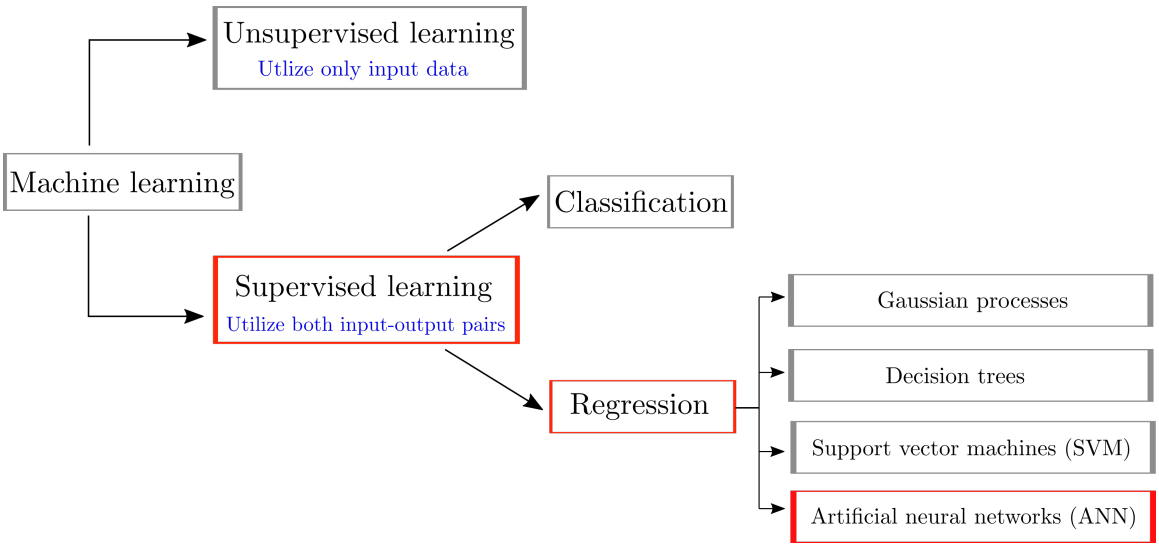


Figure 1.2: Machine learning classification.

are continuous values. As both specific cutting forces and maximum tool temperatures are continuous values a regression problem is being solved in this work.

There are several techniques available to solve a regression problem, as shown in the Figure 1.2. Artificial neural networks (ANN) are selected for this work. ANN's are especially useful for mapping complex relationships whose analytical representations are difficult. Several researchers [6], [7], [8] stated that the ANN's have the unique capability of learning through examples and generalizing the learned information. During the process of training is when neural networks acquire the knowledge of underlying relationships between independent and dependent variables. This trained neural network can be used to provide projections given new situations of interest and answer "what if" questions.

1.2.1 Artificial neural networks

An artificial neural network (ANN) is a data processing and modelling technique that arose in pursuit of mathematical modeling of the learning process based on the human brain. The studies on this subject started in 1942 with the mathematical modeling of neurons, the biological units that constitute the brain, and the appli-

cation of this model to computer systems; afterwards, it was utilized in many fields parallel to the development of computer systems. ANNs have shown to be effective as computational processors for various classification, data compression, forecasting and combination problem solving [9]. More about the modeling of ANNs, activation functions, the training algorithm, network architectures, and inputs and outputs are later presented in chapter 6.

CHAPTER 2: LITERATURE REVIEW

2.1 Finite element method for modeling machining

In recent years, the finite element method has become the main tool for simulating material removal processes. Early analyses were made by [10] and [11] who analyzed the steady state orthogonal cutting. Until the mid-1990s, most of the researchers used in-house finite element codes for modeling however, the use of commercial packages has increased recently. General-purpose FEM codes capable of modelling the machining process including Nike2D, Abaqus/Standard, Abaqus/Explicit, Marc, Algor, Ls Dyna etc.

Ceretti [12] conducted a simulation of the orthogonal plane strain cutting process using FE software Deform2D. To perform this simulation with relevant accuracy, damage criteria has been defined. Moreover, the influence of cutting parameters such as cutting speed, rake angle, and uncut chip thickness were also studied. Later, the computed cutting forces, temperatures, deformations, and the chip geometry were compared with actual cutting experiments.

Halil et al. [13] modeled orthogonal machining using various implicit and explicit finite element codes like MSC, Thirdwave AdvantEdge, and Deform2D. Both in MSC and Thirdwave AdvantEdge the separation criterion was not defined and the chip formation was assumed due to plastic flow. However, they defined CockroftLatham damage criterion in Deform2D which required specifying a predefined damage value. Finally, they concluded that the results obtained from Deform2D were in close agreement with experimental results.

2.1.1 Finite element formulations

There are 3 types of formulations available in the literature: Eulerian formulation, updated/total Lagrangian formulation and arbitrary Lagrangian formulation. Presently, total/updated Lagrangian formulations are being used widely.

2.1.1.1 Eulerian formulation

In this particular formulation the mesh is fixed in space and the material passes through the mesh to simulate the chip formation. Implementing this formulation requires the user to know the chip shape, shear angle, and contact conditions before modeling. Hence, this technique cannot be used for modeling serrated chips. Strenkowski and Carrol [14] implemented Eulerian formulation and viscoplastic material model to simulate chip formation. Childs and Mackawa [15] when implementing Eulerian formulation considered an initial straight chip shape defined entirely by shear plane angle, feed and tool geometry. The results (cutting forces) obtained by them were not in good agreement with the experimental studies.

2.1.1.2 Updated/Total Lagrangian formulation

In this formulation the elements are attached to the material. The material coordinates of nodes are time invariant that is nodes are coincident with material points. As the nodes are coincident with material points in the Lagrangian mesh, nodes on the boundaries remain on the boundary through out the problem. This reduces the complexity of imposing boundary conditions compared to the Eulerian formulation.

In total Lagrangian [16] formulation, the weak form involves integrals over reference configuration and derivatives are taken with respect to the material coordinates. Whereas in updated Lagrangian formulation, the integrals in weak form are taken with respect to the deformed configuration and the derivatives are taken with respect to spatial coordinates. Both total Lagrangian and updated Lagrangian essentially represent same mechanical behavior (Lagrangian) and can be transformed to each

other. The chip formation simulation done by Klamecki used Lagrangian formulation. This approach is more popular among the researchers because it allows chip formation from incipient. However, determining a physical chip separation criterion is still a critical research area and so far no criterion has been universally accepted. Shih [17] developed and implemented plane strain Lagrangian finite element formulation to simulate orthogonal metal cutting for continuous chip formation. He considered a material model that included elasticity, viscoplasticity, thermal effects as well as high strain and strain rate effects. His contact model had stick-slip contact formulation and presented results for stress, strain, temperature and strain rates in primary and secondary shear zones. He also compared residual stress distribution with experimental results obtained from X-ray diffraction measurements. Other researchers who used Lagrangian formulation are Sashara and Shirakashi [18], etc. Marusich and Ortiz [19] developed an updated Lagrangian model of high speed orthogonal machining where they employed continuous re-meshing and adaptive meshing to overcome the difficulties of element distortion.

2.1.1.3 Arbitrary Lagrangian-Eulerian Formulation

The Eulerian and Lagrangian formulations have their own advantages and disadvantages. Therefore a hybrid technique, Arbitrary Lagrangian Eulerian (ALE) formulation, which combines the advantages of both Eulerian and Lagrangian method has been developed [16]. In this method user can define a part of the mesh to have Lagrangian formulation and a part can have Eulerian formulation, such that advantages of both the methods are utilized. In machining simulations with ALE formulation, the boundary nodes and the nodes at the interface locations remains coincident with the material points and hence a Lagrangian formulation is considered for them. Where as the internal nodes are modeled with Eulerian formulation in order to overcome severe element distortion in primary and secondary shear zones. Olovsson et al. [20] developed one of the first ALE models for simulation of orthogonal cutting. They con-

sidered an elastic-plastic material model with isotropic hardening and used Coulomb's friction model to define the contact between chip and tool.

2.1.2 Constitutive models

The constitutive models for machining simulations should adequately represent the behavior of the material ,under extreme conditions encountered, in machining process. Chip formation is the result of material plastic deformation during relative motion between the tool and the workpiece. One of the first and widely used constitutive equation that expresses the flow stresses of the workpiece material as a function of strain, strain rate and temperature is the Johnson-Cook model [21], [22], [23]. It is defined by

$$\sigma(\bar{\epsilon}, \dot{\bar{\epsilon}}, T) = (A + B\bar{\epsilon}^n) \left[1 + C\, ln\Big(\frac{\dot{\bar{\epsilon}}}{\dot{\epsilon}_0}\Big)\right] \left[1 - \bar{T}^m\right] \tag{2.1}$$

Here A is the yield stress, B is the strength coefficient, C is the strain rate constant, T is the temperature, σ is the flow stress, n is the strain-hardening exponent and m is the thermal softening exponent. These parameters need to be determined experimentally. All these parameters are determined mostly by impact compression tests at moderate deformation rates. Usui et al. [10] used a split Hopkinson bar apparatus to obtain the deformation characteristics of different types of steel samples for simulating the high strain rates and temperature effects encountered in machining. The samples were deformed under high-speed compression tests by the use of the apparatus, and as a result, strains of up to 2.0 and strain rates of up to 2000 s^{-1} were obtained.

2.1.3 Contact modeling

One of the common and basic friction models is the one developed by Coulomb. This model was used in the machining model of O.Pantale et al. [21], Maurisch and Ortiz [19]. According to this model the bodies under contact are assumed to stick together if

$$||T_t|| < \mu |T_n| \tag{2.2}$$

and in a relative motion

$$||T_t|| = \mu|T_n| \tag{2.3}$$

where μ is the co-efficient of friction, T_t is the tangential component of surface traction and T_n is the normal force acting on the segment. The other friction model which is commonly used in the recent times is the Zorev's model.

2.1.4 Material Separation

To make a reliable FEM simulation in Lagrangian formulation, an appropriate separation criterion is necessary. A good criterion must reflect the mechanics and the physical mechanism of the material and produce reasonable results, such as chip geometry, cutting forces, temperature distribution and residual stresses [24]. The chip separation criterion in modeling machining can be done in two ways: geometrical and physical. The main disadvantage of geometric based criterion is that, it is not based on the physics and mechanics of the chip formation. Ideally, in machining there is no physical gap between the crack and tool tip. Hence, if critical distance criterion is employed, the minimum threshold value for node separation should be zero or extremely small. On the other hand physical chip separation criterion based on some physical quantity is more appropriate than using geometrical criterion.

2.2 Artificial neural network modeling

In recent years, few studies have been reported in the literature involving the use of ANNs for engineering applications. Ovali et al. [25] conducted a study on predicting forces of austempered grey iron using ANN and concluded that artificial neural networks have more ability than regression analysis to solve problems having non-linear relationships. Kara et al. [26] also performed modeling of cutting forces during the orthogonal machining of AISI 316L stainless steel with cutting speed, feed rate and coating type as the input parameters using both multiple regression and ANN and concluded that results obtained from ANN are good(low error) in terms of

prediction. Asokan et al. [27] and Al-Ahmari [28] also compared regression analysis with ANN and concluded that ANNs are better(low error) in terms of performance.

Kumar and Singh [29] used an ANN based model to predict stability in turning, and applied tangent sigmoid activation function for model training. Markopoulos et al. [9] proposed ANN models, developed using Matlab and Netlab tools, for predicting surface roughness in electrical discharge machining. They used hyperbolic tangent sigmoid activation function (Tanh) between input and hidden layers and used linear activation function between the last hidden layer and the output layer. One possible explanation for this approach was for easy computation of the derivative taken at the cost function at the last layer.

Rajkumar et al. [30] investigated the training data and transfer functions required to produce an efficient neural network architecture for predicting aerodynamic coefficients by using linear, hyperbolic tangent and sigmoid activation functions and used Levenberg-Marquardt as training algorithm. They concluded that the 3 layered neural network with sigmoid activation function produced least error in prediction. Dahbi et al. [31] in his research work stated that a feed forward ANN with back propagation algorithm gave the accurate results and found the results to be in good agreement with the experimental data. He developed 8 neural networks with different network architectures and predicted 8 outputs with 3 inputs using MATLAB neural network tool box.

Cherukuri et al. [32] used an artificial neural network to model turning stability and observed that the number and distribution of training points influenced the ability of the ANN model to capture the smaller, more closely spaced lobes that occur at lower spindle speeds. They concluded that Deep and narrow neural networks performance is found to be more accurate than shallow and wide networks. They have also examined the sensitivity of the ANN performance to its architecture, and concluded that the performance of the ANN is closely linked to the selected number of hidden

layers and neurons per layer. Ihsan et al. in [33] applied regression analysis and artificial neural network in modeling tool chip interface temperature in machining. They used Levenberg-Marquardt (LM) algorithm for training the networks and concluded that the tool-chip interface temperature temperature equation derived from regression analysis and ANN model can be used for prediction.

Authors Abdullah et al. [34] and Sakir Tasdemir in [35],[36] determined the best neural network architecture by monitoring statistical results obtained by computing mean sqaured error (MSE) and coefficient of determination (R^2). The model with least MSE and highest R^2 was selected to be suitable network architecture, similar approach was used in this work.

Regarding the activation functions used in ANN modeling, Pontes et al. [37] in their work stated that, 11 publications that used hyperbolic tangent activation functions and 7 publications that used sigmoid activation function. Correa et al. [38] in his work declared that there are no standard algorithms for choosing the network parameters (number of hidden layers, number of nodes in the hidden layers, and the activation functions). Haykin [6] in his work stated that hyperbolic tangent activation will lead to faster convergence in training due to its symmetrical shape. He also added that there are no standard methods to determine the number of hidden layers and neurons.

CHAPTER 3: FINITE ELEMENT MODELING

Modeling and simulations are a substitute for physical experimentations, where problems are solved by utilizing the computational ability of the computers. We define a mathematical model that contains all the features of a physical model and represent it in the virtual format. Proper modeling techniques improve the computational efficiency, and at the same time ensure the reliability and accuracy of the results. A universal computer modeling concept GIGO (garbage in , garbage out) implies bad input will result in bad output. Computers operate on strict logic and invalid inputs are going to produce unrecognizable outputs (garbage). Therefore, providing accurate inputs to the model is considered highly important.

Modeling machining is considered to be a challenging activity [39]. First of all, the strain rates observed are very high. This holds even for low cutting speeds. There is no unified and generally accepted theory regarding the exact chip formation mechanism, which is mainly due to the phenomenon taking place in the deformed regions. Additionally, the temperature rise in these regions due to plastic deformation and friction induce material softening and alter the workpiece material properties with respect to strain rates and temperatures. Moreover, data for the workpiece material at varying temperatures and strain rates during machining is not easily found in the literature. On top of this, machining has three main sources of non-linearity: geometric, material and contact. The rise in temperature should also be taken into account, which indicates that a coupled analysis must be carried out.

3.1 FEA Software Package

Currently, in the market, there are many commercial software packages available for solving various engineering problems. The usage of different software packages will have different capabilities. This motivates us to choose the software package that is widely used by researchers and the one which has promising results. Abaqus has two solvers: the one which works on the implicit time integration scheme known as Abaqus/Standard, and the other that works on the explicit time integration scheme known as Abaqus/Explicit. The first one has the capabilities to perform a wide range of linear, non-linear problems in the domain of static, dynamic and thermal analysis. On the other hand, Abaqus/Explicit is suitable for modeling transient dynamic events and is very efficient for highly non-linear problems that involve challenging contact conditions. Ali et. al [40] compared the results of 4 different finite element software packages i.e AdvantEdge, Abaqus/Explicit, Deform2D and Forg to simulate machining process of Titanium alloy Ti-6Al-4V. The comparision concluded that the results obtained by Abaqus/Explicit are more accurate. FJ Harewood et. al [41] has also conducted a study using both standard and explicit solvers and concluded that explicit solver is better suited to deal with complex contact and sliding conditions, particularly in the case of large element deformation. Hence, Abaqus/Explicit solver is used in this research.

Figure 3.1 shows the flow chart employed by Abaqus while solving a problem. Abaqus/CAE is utilized to perform pre-processing. It is divided into multiple modules, where each module follows a logical aspect of the modeling process which involves defining the model geometry, boundary conditions, mesh, material properties, and other modeling parameters. After pre-processing Abaqus/CAE generates an input file (.inp extension) which is sent to the solver. The results are written into an output file, which is utilized for post-processing, which can be performed using Abaqus/CAE or other programming languages like Python or MATLAB.

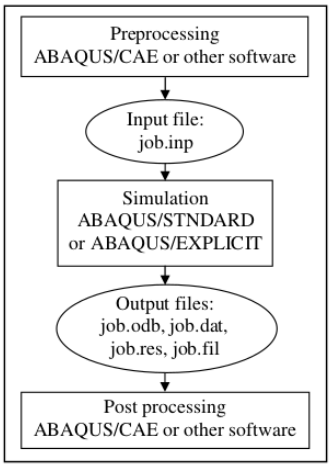


Figure 3.1: Steps involved in Abaqus [3].

The components that are required for building a finite element model for orthogonal machining are as follows:

- Model formulation
- Finite element model set up
- Material modeling
- Contact modeling
- Damage modeling

3.2 Model Formulation

Simulating orthogonal machining involves solving a fully coupled thermo-visco-plastic problem. Here, the updated Lagrangian formulation is used. This formulation is considered most efficient for many applications in the area solid mechanics [16]. The key equation discretized is the momentum equation, which is expressed in terms of the Eulerian (spatial) coordinates and Cauchy (physical) stress.

3.2.1 Strong form and weak form

The governing equations for the updated Lagrangian formulation to represent the mechanical behavior of a continuous body are given by the strong form, or generalized

momentum balance, which consists the following set of equations:

Momentum equation:

$$\frac{\partial \sigma_{ji}}{\partial x_j} + \rho b_i = \rho \dot{v_i} \tag{3.1}$$

where σ_{ji} is the stress tensor, ρ is the density, ρb_i is the body force per unit volume, and $\rho \dot{v_i}$ is the acceleration

Traction boundary conditions:

$$n_j \sigma_{ji} = t_i \;\; on \;\; \partial\Omega_t \quad and \quad v_i = \bar{v_i} \;\; on \;\; \partial\Omega_v \tag{3.2}$$

t_i and v_i are the prescribed traction and velocity on the surfaces $\partial\Omega_t$ and $\partial\Omega_v$, respectively. The principal of virtual power is the weak form for both the set of equations (Equation 3.1 and Equation 3.2) mentioned above. The weak form is obtained by multiplying the strong form (Equation 3.1) by the test function δv_i, also known as virtual velocity and by following a series of mathematical operations which involves substituting the traction boundary conditions, applying Gauss's theorem, and integration by parts. The equation obtained is:

$$\int_\Omega \frac{\partial(\delta v_i)}{\partial x_j} \sigma_{ji} d\Omega + \int_\Omega \delta v_i \rho \dot{v_i} d\Omega = \int_\Omega \delta v_i \rho b_i d\Omega + \int_{\partial\Omega_t} \delta v_i \bar{t_i} \partial\Omega_s \tag{3.3}$$

where Ω represents the domain under consideration and $\Omega_s = \partial\Omega_t \cup \partial\Omega_v$. The first expression on the left side of the equation 3.3 represents the virtual internal power (δP^{int}), also known as virtual stress power, and second expression is virtual kinetic power (δP^{kin}). The entire right side of the equation 3.3 represents the virtual external power (δP^{ext}). Hence, the equation can be simplified as,

$$\delta P^{int} + \delta P^{kin} = \delta P^{ext} \tag{3.4}$$

3.2.2 Application of finite element method

According to the finite element method the current domain Ω is discretized into Ω_e finite elements, then the position $x_i(X,t)$, velocity $v_i(X,t)$ and acceleration $a_i(X,t)$

for the elements are expressed as,

$$x_i(X,t) = N_I(X)x_{iI}(t) \tag{3.5}$$

$$v_i(X,t) = N_I(X)v_{iI}(t) \tag{3.6}$$

$$a_i(X,t) = N_I(X)\dot{v}_{iI}(t) \tag{3.7}$$

Here, $N(X)$ stands for shape function in material coordinate system that can be mapped to spatial coordinates, I represents the node number and i corresponds to the components such that $i = 2$ for 2D space and $i = 3$ for 3D space. Thus, the terms in Equation 3.3 can we rewritten by using shape functions as,

$$\int_\Omega \frac{\partial N_i}{\partial x_j}\sigma_{ji}d\Omega + \dot{v}_{iJ}\int_\Omega \rho N_I N_J d\Omega = \int_\Omega N_I \rho b_i d\Omega + \int_{\partial\Omega_t} N_I \bar{t}_i d\Omega_s \tag{3.8}$$

where the complete right hand side of the equation in known as external nodal force denoted as f_{iI}^{ext}, given by

$$f_{iI}^{ext} = \int_\Omega N_I \rho b_i d\Omega + \int_{\partial\Omega_t} N_I \bar{t}_i d\Omega_s \tag{3.9}$$

The first expression on the left hand side is known internal nodal force denoted as f_{iI}^{int}, given by

$$f_{iI}^{int} = \int_\Omega \frac{\partial N_i}{\partial x_j}\sigma_{ji}d\Omega = \int_\Omega B_{Ij}\sigma_{ji}d\Omega \tag{3.10}$$

where, B_{Ij} is a matrix with elements of shape function derivatives, given by

$$B_{jI} = \frac{\partial N_i}{\partial x_j} \tag{3.11}$$

The second expression on the left hand side is known as kinetic (or inertial) nodal force denoted by f_{iI}^{kin}, given by

$$f_{iI}^{kin} = \dot{v}_{iJ}\int_\Omega \rho N_I N_J d\Omega \tag{3.12}$$

The Equation 3.12 can be represented as a product of mass matrix and nodal acceleration, where the mass matrix denoted by M_{ijIJ}, given by

$$M_{ijIJ} = \delta_{ij}\int_\Omega \rho N_I N_J d\Omega \tag{3.13}$$

Thus, the kinetic nodal force can we rewritten as,

$$f_{iI}^{kin} = \dot{v}_{iJ} \int_{\Omega} \rho N_I N_J d\Omega \tag{3.14}$$

Finally, the Equation 3.8 , can be written as, which also

$$M_{ijIJ}\dot{v}_{ij} + f_{iI}^{int} = f_{iI}^{ext} \tag{3.15}$$

This equation is known as Equation of motion.

3.2.3 Finite element equation for transient thermal problem

The semi - discrete finite element equation for heat transfer is given by

$$C_{ij}\dot{\theta}_j + K_{ij}\theta_j = q_i \tag{3.16}$$

where C represents the heat capacity matrix, K represents the heat conductivity matrix and q represents the heat generation source.

3.2.4 Explicit solver

The explicit dynamic analysis uses a central difference scheme [3], given by the equation.

$$\dot{u}^n_{i+\frac{1}{2}} = \dot{u}^n_{i-\frac{1}{2}} + \frac{\Delta t_{i+1} + \Delta t_i}{2}\ddot{u}^n_i \tag{3.17}$$

$$u^n_{i+1} = u^n_i + \Delta t_{i+1}\dot{u}^n_{i+\frac{1}{2}} \tag{3.18}$$

where u^n is the displacement or rotational degree of freedom of node n. And the subscript i denotes the increment number of the explicit step. $\dot{u}$ and $\ddot{u}$ represent the velocity and acceleration respectively. At the beginning of the increment the accelerations are calculated using equation 3.19

$$\ddot{u}^n_i = M^{-1} \cdot (F_i - L_i) \tag{3.19}$$

where M is the diagonal lumped mass matrix, F is the external load vector, and L is the internal force vector. The acceleration is substituted in Equations (3.17) and (3.18) to compute the velocities and displacements respectively. From the above

equations it can be seen that the explicit analysis uses the values from the previous increments to advance the computations, thus requiring no iterations and it uses a diagonal mass matrix increasing the computational efficiency of the procedure [3].

The main drawback of the explicit analysis is that, it integrates through time by using many small time increments. The central difference operator used in this analysis is conditionally stable which requires small time increments for accurate results. The stable time increments for the analysis is given by the equation (3.20)

$$\Delta t \leq \frac{2}{\omega_{max}} \tag{3.20}$$

where ω_{max} is the highest frequency of the system. In Abaqus, the stable time increments given in equation (3.20) is approximated as the smallest transit time for the dilatational wave to travel across the smallest element in the mesh.

$$\Delta t \approx \frac{L_{min}}{c_d} \tag{3.21}$$

where L_{min} is the length of the smallest element in the mesh and c_d is the dilatational wave speed given by

$$c_d = \sqrt{\frac{\lambda + 2\mu}{\rho}} \tag{3.22}$$

where ρ is the density of the material and λ, μ are the Lame's constants given in terms of Young's modulus E and Poisson's ratio ν, as shown in equation (3.23) and (3.24) respectively.

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \tag{3.23}$$

$$\mu = \frac{E}{2(1+\nu)} \tag{3.24}$$

Thus for a very fine mesh, the stable time increment is very small leading to large computational times. And also a dynamic analysis requires large amount of time to achieve steady state, whereas an equilibrium is achieved with a few increments in a static analysis.

3.3 Finite element model set up

The material properties for both the workpiece (Aluminum alloy-A2024-T351) and the cutting tool (Tungsten carbide - WC) are shown in Table 3.1. The geometry of the workpiece and the tool are similar to the one considered by Mabrouki et al. in [42]. Figure 3.2 presents the finite element model setup of this research work. Four node quadrilateral elements, CPE4RT, and three node triangular elements, CPE3T, with reduced integration and plane strain formulation were used for meshing both the workpiece and tool. The total number of nodes and elements used for meshing are 22794 and 22447 respectively.

The nodes along the length and breadth of the workpiece were fully constrained in x and y directions, whereas the tool (all the nodes) is constrained in y direction and given a velocity V_c in negative x direction as shown in the Figure 3.2. The clearance angle and the tool nose radius are fixed to be 7° and 20 µm.

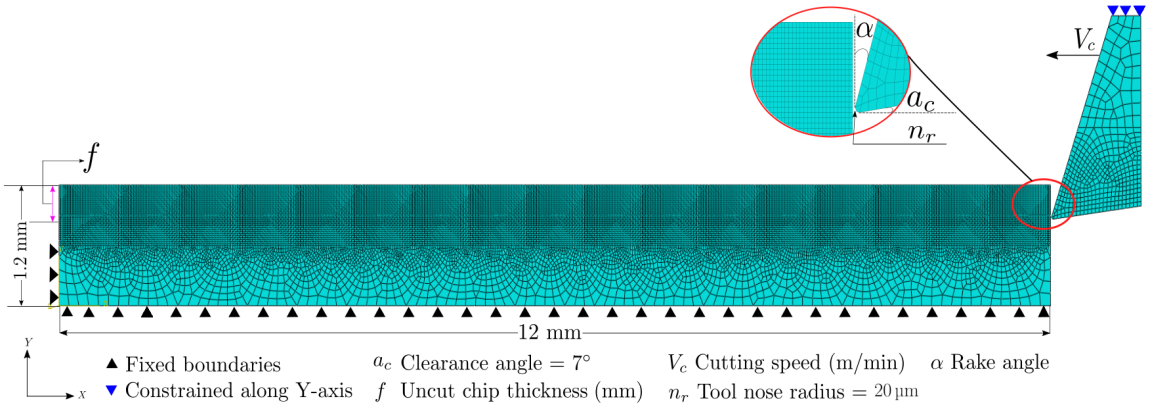


Figure 3.2: Finite element model setup.

3.4 Material modeling

Johnson-Cook (JC) constitutive model was used for material modeling. The model is formulated empirically and it is based on Mises plasticity, where Mises yield surface (J2 plasticity theory) is associated with the flow rule. JC constitutive equation considers isotropic hardening and is capable of modeling the thermo-visco-plastic problem

Table 3.1: Material properties of workpiece and tool.

Physical property	Workpiece (Al 2024-T351)	Tool (WC)
Density, ρ (Kg/m^3)	2700	11900
Young's Modulus, E (Gpa)	73	534
Poisson's ratio, ν	0.33	0.22
Specific heat, (J/Kg/°C)	$C_p = 0.557$ T+877.6	400
Thermal expansion coeff., α_d (°C^{-1})	$\alpha = (8.9e^{-3}$ T + $22.6)e^{-6}$	NA
Thermal conductivity, (W/(m-°C))	for: $25 \leq$ T $<$ 300	
	$\lambda = 0.247$T + 114.4	50
	for: $300 \leq$ T $\leq$ T$_{melt}$	
	$\lambda = 0.125$T + 226	50

over a strain rate range of 10^2 to 10^5 s^{-1}. The flow stress is represented as a function of strain, strain rate, and temperature (see Equation 3.25). The first term in the equation accounts for isotropic hardening. The second term accounts for strain rate hardening, and the third term accounts for thermal softening. The material parameters A, B, n, C and m for the JC model (see Table 3.2) are obtained from [43] and [42].

$$\sigma(\bar{\epsilon}, \dot{\bar{\epsilon}}, T) = (A + B\bar{\epsilon}^n) \left[1 + C\, ln\Big(\frac{\dot{\bar{\epsilon}}}{\dot{\epsilon}_0}\Big)\right] \left[1 - \bar{T}^m\right] \tag{3.25}$$

Here, $\bar{T}$ in Equation 3.25 is given by:

$$\bar{T} = \begin{cases} 0, \ T < T_{trans}, \\ \dfrac{T - T_{trans}}{T_{melt} - T_{trans}}, \ T_{trans} < T < T_{melt}, \\ 1, \ T > T_{melt} \end{cases}$$

Table 3.2: Johnson-Cook model parameters for Al2024-T351.

A $(M\,Pa)$	B $(M\,Pa)$	n	C	m	$T_{transition}$ $(°C)$	T_{melt} $(°C)$
352	440	0.42	0.0083	1	25	520

3.5 Contact Modeling

Contact modeling in the secondary deformation zone, at the interface of the chip and the rake face of the tool is of great importance. From experimental results, it has been found and verified that two contact regions may be distinguished in dry machining, the sticking region, and the slipping region [39]. Zorev proposed a friction model in [44], where he showed that the normal stress (σ_n) in the secondary deformation zone is maximum at the tool tip and reduces to zero at a point where the chip loses contact from the rake face, as shown in Figure 3.3.

Zorev's model was widely used in the research community to model friction at the tool-chip interface. However, in slip zone (l_{slip}) the coefficient of friction (μ) is assumed to be constant and independent of σ_n [45] . Simple Coulomb's friction model with an average μ is used due to its simplicity. Such an approach for contact modeling in machining has been criticized and found to be misleading [4]. It is worth noting that if the coefficient of friction is constant over the tool rake face, as per the Coulomb's model, the curve for shear stress (τ) and σ_n in Figure 3.3 should be parallel, but it is not the case. Hence, the average coefficient of friction is no longer able to characterize the relationship between σ_n and τ at the tool-chip interface accurately.

$$\tau = \sum_{m=1}^{m=p} a_m \sigma_n^m \tag{3.26}$$

Therefore, sticking region in this work is modeled by the stress based friction model proposed by Yang and Liu [46]. Patel et al. [4] based on the model proposed by Yang and Liu generated a stress based polynomial model by fitting the data obtained from [47] to a polynomial of degree p=3 as per the Equation 3.26, and determined a relationship between τ and σ_n given by the Equation 3.28. Hence, the sticking region is modeled by using Equation 3.27 and the slipping region is modeled by Equation

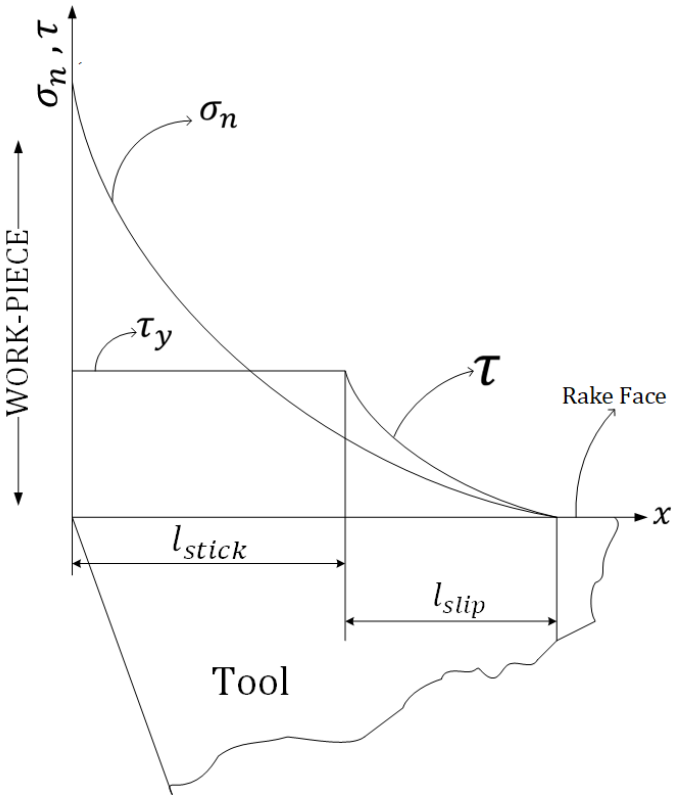


Figure 3.3: Normal stress $\sigma_n(x)$ and shear stress τ distribution at the tool rake face as per Zorev's friction model [4].

3.28 where the limiting shear stress (τ_y) is 203 MPa.

$$\tau = \tau_y, \quad \mu\sigma_n \geq \tau_y \tag{3.27}$$

$$\tau = 2.795e^{-6}{\sigma^3}_n - 0.003285{\sigma^2}_n + 1.372\sigma_n, \quad \mu\sigma_n \leq \tau_y \tag{3.28}$$

In Abaqus, the tool-chip interaction was modeled using the penalty stiffness contact formulation where the tool was considered as master surface and the chip was considered as slave surface. In addition, the self contact of the chip was also defined using penalty contact formulation.

3.6 Damage modeling

Chip formation takes place as a result of damage and fracture of a material due to the action of the cutting tool. Finite element simulations require a criterion to

simulate chip separation from the bulk when the tool moves and interacts with the workpiece. The chip separation criterion should reflect closely the physics and mechanics of chip formation to achieve reliable results. In this work, the Johnson-Cook damage model [48] was used to model machining as a process resulting from damage and fracture in a material. According to this model the overall damage in a material occurs in two steps [3] :

- Damage initiation
- Damage evolution

Damage initiates in a material when the damage parameter, ω, defined as:

$$\omega = \sum \frac{\Delta\bar{\epsilon}}{\bar{\epsilon}_d} \tag{3.29}$$

exceeds or equals one. The numerator, $\Delta\bar{\epsilon}$, is the increment in equivalent plastic strain, whereas the denominator, $\bar{\epsilon}_d$, is equivalent plastic strain at the onset of damage initiation and is given by the Equation 3.30. The parameters D_1 to D_5 are shown in the Table 3.3. The parameter D_5 is zero which indicates that the temperature does not have any affect on the damage of Aluminum [42].

$$\bar{\epsilon}_d = \left[D_1 + D_2 exp\left(D_3 \frac{p}{\bar{\sigma}}\right)\right]\left[1 + D_4 ln\left(\frac{\dot{\bar{\epsilon}}}{\dot{\epsilon}_0}\right)\right]\left[1 + D_5 \bar{T}\right] \tag{3.30}$$

Table 3.3: Johnson-Cook damage model parameters for Al2024-T351.

D_1	D_2	D_3	D_4	D_5
0.13	0.13	1.5	0.011	0

Figure 3.4 shows the stress-strain curve for a material undergoing damage. The material response is linear from the point a to b followed by isotropic hardening and inelastic deformation from b to c. At point c, the damage initiation criterion is satisfied (i.e., $\omega \geq 1$) where, $\bar{\epsilon}_d$ and $\bar{\sigma}_d$ are the equivalent plastic strain and yield stress,

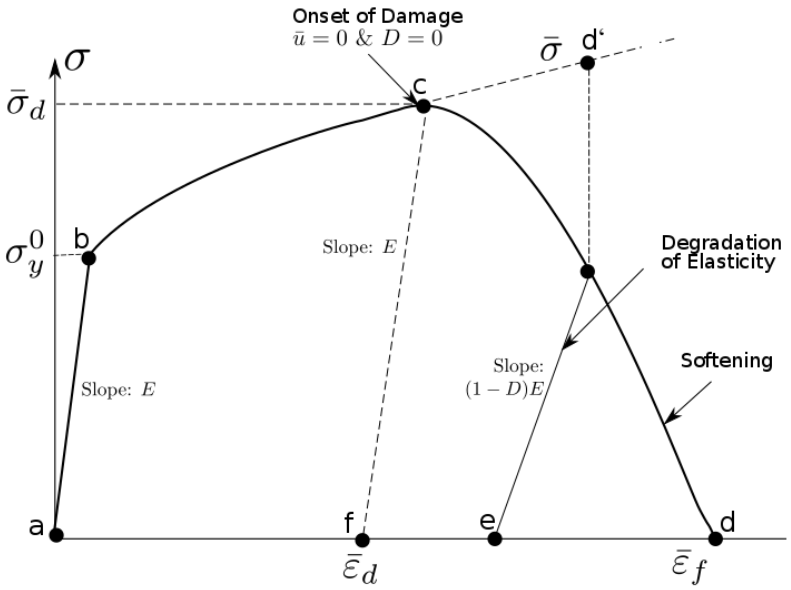


Figure 3.4: Stress-strain curve for material undergoing damage.

respectively. Beyond point c the damage manifests itself into two forms: softening of the yield stress and degradation of the elasticity due to which the load carrying capacity of the material is remarkably decreased. In the context of damage mechanics c to d can be viewed as the degraded response of the curve c to d', that the material would have followed in the absence of damage [3]. D is the overall damage variable defined such that $D = 0$ at the onset of damage and when $D = 1$ the stiffness of the element is completely degraded. The equivalent plastic strain at this point is denoted by $\bar{\epsilon}_f$. Once the damage initiation criterion is satisfied, the material stiffness degrades progressively according to the specified damage evolution model, eventually leading to the complete damage of the material.

There are two forms of damage evolution: linear evolution and exponential evolution. According to linear evolution the overall damage variable D is defined as:

$$D = \frac{\dot{\bar{u}}\bar{\sigma}_y}{2G_f} \tag{3.31}$$

where, $\dot{\bar{u}}$ is rate of equivalent plastic displacement , G_f is critical energy release rate and $\bar{\sigma}_y$ is yield stress after the onset of damage. When $D = 1$, in an element, the

element is considered to be completely degraded and it is removed from the model.

According to exponential evolution, the overall damage variable D is defined as:

$$D = 1 - exp\left[- \int_0^{\bar{u}} \frac{\bar{\sigma}_y d\bar{u}}{G_f} \right] \tag{3.32}$$

Since D approaches 1 when $\bar{u}$ approaches infinity, in Abaqus, D is taken to be one when the total dissipated energy for each element approaches $0.99G_f$.

In this work, exponential evolution was defined across the area of uncut chip thickness (see Figure 3.5) and linear evolution was defined for the remaining area (see Figure 3.5). This approach and the values of G_f were adopted from the work done by Patel & Cherukuri in [49].

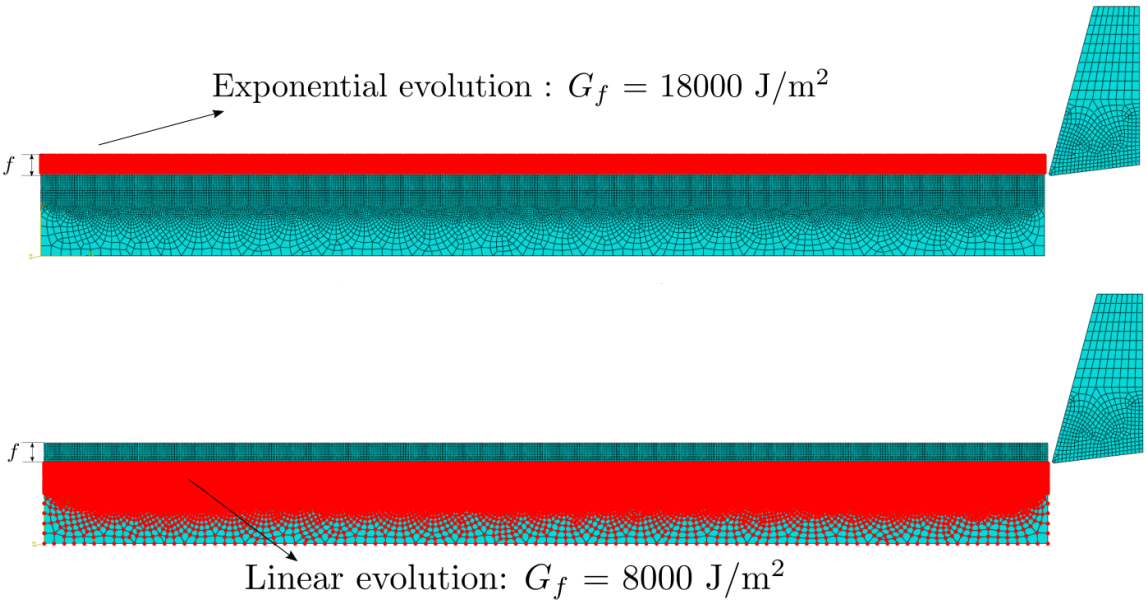


Figure 3.5: Exponential and linear evolution.

3.7 FEA simulations and data extraction

Parametric studies were performed on the developed finite element model for seven rake angles, four uncut chip thickness values and seven cutting speeds (see Table 3.4) resulting in total 196 (7 x 4 x 7) simulations. The nose radius and clearance angle were constant throughout the simulations. The results were captured for every 1E^{-06}s . The total run time for all the simulations was 2352 hours with each simulation taking

about 12 hours on average.

Table 3.4: Parameters for simulations.

Rake angle (deg)	Uncut chip thickness (mm)	Cutting speed (m/min)
-3	0.1	100
0	0.2	200
5	0.3	400
8	0.4	600
15		800
17.5		1000
20		1200

The other important and time consuming process was extracting the required data from the output files. Loading all the output files into the module and manually extracting the data sets is exhaustive and can create scope for making errors. Hence, the process of extracting the data sets was automated, where a python script was developed which loaded the output files into the Abaqus kernel and calculated the specific cutting forces and maximum tool temperatures. Later, the calculated values were stored into an XLSX file using the same script.

The data extracted from finite element simulations is arranged in the Tables 3.5, 3.6, 3.7 and 3.8.

Table 3.5: Part1 - Data obtained from finite element simulations.

S.No.	Input parameters			Outputs	
	Rake angle (deg)	Uncut chip thickness (mm)	Cutting speed (m/min)	Specific cutting force (N/mm^2)	Maximum tool temperature (°C)
1	-3	0.1	100	874	164
2	-3	0.1	200	895	184
3	-3	0.1	400	945	208
4	-3	0.1	600	890	221
5	-3	0.1	800	931	228
6	-3	0.1	1000	848	233
7	-3	0.1	1200	922	237
8	-3	0.2	100	733	182
9	-3	0.2	200	750	199
10	-3	0.2	400	757	217
11	-3	0.2	600	758	227
12	-3	0.2	800	780	232
13	-3	0.2	1000	755	236
14	-3	0.2	1200	782	242
15	-3	0.3	100	675	189
16	-3	0.3	200	687	209
17	-3	0.3	400	693	231
18	-3	0.3	600	694	244
19	-3	0.3	800	695	245
20	-3	0.3	1000	687	259
21	-3	0.3	1200	701	261
22	-3	0.4	100	700	191
23	-3	0.4	200	709	206
24	-3	0.4	400	707	214
25	-3	0.4	600	708	236
26	-3	0.4	800	705	230
27	-3	0.4	1000	683	262
28	-3	0.4	1200	684	264
29	0	0.1	100	859	161
30	0	0.1	200	875	181
31	0	0.1	400	911	204
32	0	0.1	600	743	216
33	0	0.1	800	859	225
34	0	0.1	1000	872	230
35	0	0.1	1200	963	234
36	0	0.2	100	705	179
37	0	0.2	200	721	193
38	0	0.2	400	738	215
39	0	0.2	600	738	220
40	0	0.2	800	735	225
41	0	0.2	1000	750	230
42	0	0.2	1200	767	234
43	0	0.3	100	658	175
44	0	0.3	200	660	203
45	0	0.3	400	668	223
46	0	0.3	600	668	231
47	0	0.3	800	669	240
48	0	0.3	1000	667	246
49	0	0.3	1200	678	249
50	0	0.4	100	681	188
51	0	0.4	200	680	202
52	0	0.4	400	657	219

Table 3.6: Part2 -Data obtained from finite element simulations.

S.No.	Input parameters			Outputs	
	Rake angle (deg)	Uncut chip thickness (mm)	Cutting speed (m/min)	Specific cutting force (N/mm^2)	Maximum tool temperature (°C)
53	0	0.4	600	643	234
54	0	0.4	800	678	233
55	0	0.4	1000	709	245
56	0	0.4	1200	626	244
57	5	0.1	100	806	156
58	5	0.1	200	817	177
59	5	0.1	400	791	199
60	5	0.1	600	836	199
61	5	0.1	800	916	218
62	5	0.1	1000	820	222
63	5	0.1	1200	707	227
64	5	0.2	100	664	169
65	5	0.2	200	677	183
66	5	0.2	400	696	199
67	5	0.2	600	700	206
68	5	0.2	800	724	209
69	5	0.2	1000	700	212
70	5	0.2	1200	668	216
71	5	0.3	100	618	180
72	5	0.3	200	651	208
73	5	0.3	400	631	211
74	5	0.3	600	631	225
75	5	0.3	800	639	229
76	5	0.3	1000	632	234
77	5	0.3	1200	629	239
78	5	0.4	100	637	179
79	5	0.4	200	587	205
80	5	0.4	400	579	220
81	5	0.4	600	601	236
82	5	0.4	800	596	235
83	5	0.4	1000	599	254
84	5	0.4	1200	599	259
85	8	0.1	100	758	150
86	8	0.1	200	780	174
87	8	0.1	400	795	196
88	8	0.1	600	821	206
89	8	0.1	800	755	212
90	8	0.1	1000	787	216
91	8	0.1	1200	724	218
92	8	0.2	100	637	166
93	8	0.2	200	647	178
94	8	0.2	400	644	192
95	8	0.2	600	636	200
96	8	0.2	800	646	204
97	8	0.2	1000	671	207
98	8	0.2	1200	692	210
99	8	0.3	100	594	174
100	8	0.3	200	605	190
101	8	0.3	400	605	206
102	8	0.3	600	610	216
103	8	0.3	800	623	222
104	8	0.3	1000	619	227

Table 3.7: Part3 -Data obtained from finite element simulations.

S.No.	Input parameters			Outputs	
	Rake angle (deg)	Uncut chip thickness (mm)	Cutting speed (m/min)	Specific cutting force (N/mm^2)	Maximum tool temperature (°C)
105	8	0.3	1200	593	233
106	8	0.4	100	560	172
107	8	0.4	200	567	187
108	8	0.4	400	563	219
109	8	0.4	600	573	233
110	8	0.4	800	581	241
111	8	0.4	1000	567	242
112	8	0.4	1200	597	247
113	15	0.1	100	703	146
114	15	0.1	200	699	169
115	15	0.1	400	656	190
116	15	0.1	600	709	199
117	15	0.1	800	595	204
118	15	0.1	1000	700	208
119	15	0.1	1200	733	214
120	15	0.2	100	583	155
121	15	0.2	200	596	171
122	15	0.2	400	601	185
123	15	0.2	600	616	195
124	15	0.2	800	598	200
125	15	0.2	1000	600	205
126	15	0.2	1200	552	210
127	15	0.3	100	547	160
128	15	0.3	200	558	175
129	15	0.3	400	559	195
130	15	0.3	600	571	206
131	15	0.3	800	556	211
132	15	0.3	1000	561	217
133	15	0.3	1200	511	223
134	15	0.4	100	515	172
135	15	0.4	200	518	189
136	15	0.4	400	524	205
137	15	0.4	600	532	213
138	15	0.4	800	527	218
139	15	0.4	1000	527	226
140	15	0.4	1200	506	229
141	17.5	0.1	100	678	144
142	17.5	0.1	200	675	167
143	17.5	0.1	400	677	186
144	17.5	0.1	600	683	203
145	17.5	0.1	800	680	201
146	17.5	0.1	1000	698	208
147	17.5	0.1	1200	794	215
148	17.5	0.2	100	565	153
149	17.5	0.2	200	572	167
150	17.5	0.2	400	581	184
151	17.5	0.2	600	582	192
152	17.5	0.2	800	587	207
153	17.5	0.2	1000	581	206
154	17.5	0.2	1200	556	213
155	17.5	0.3	100	526	158
156	17.5	0.3	200	540	170

Table 3.8: Part4 - Data obtained from finite element simulations.

S.No.	Input parameters			Outputs	
	Rake angle (deg)	Uncut chip thickness (mm)	Cutting speed (m/min)	Specific cutting force (N/mm^2)	Maximum tool temperature (°C)
157	17.5	0.3	400	544	182
158	17.5	0.3	600	539	205
159	17.5	0.3	800	542	211
160	17.5	0.3	1000	540	215
161	17.5	0.3	1200	494	219
162	17.5	0.4	100	501	166
163	17.5	0.4	200	501	191
164	17.5	0.4	400	496	201
165	17.5	0.4	600	512	211
166	17.5	0.4	800	502	222
167	17.5	0.4	1000	505	222
168	17.5	0.4	1200	458	225
169	20	0.1	100	648	143
170	20	0.1	200	635	166
171	20	0.1	400	648	185
172	20	0.1	600	697	193
173	20	0.1	800	724	207
174	20	0.1	1000	725	201
175	20	0.1	1200	702	222
176	20	0.2	100	546	151
177	20	0.2	200	549	165
178	20	0.2	400	562	182
179	20	0.2	600	579	194
180	20	0.2	800	566	206
181	20	0.2	1000	565	209
182	20	0.2	1200	539	218
183	20	0.3	100	509	155
184	20	0.3	200	516	166
185	20	0.3	400	518	180
186	20	0.3	600	527	198
187	20	0.3	800	514	214
188	20	0.3	1000	518	220
189	20	0.3	1200	530	228
190	20	0.4	100	492	158
191	20	0.4	200	486	180
192	20	0.4	400	483	198
193	20	0.4	600	502	210
194	20	0.4	800	470	212
195	20	0.4	1000	453	226
196	20	0.4	1200	495	232

CHAPTER 4: FINITE ELEMENT MODEL VALIDATION

In this chapter, the developed finite element model was validated with the experimental results published in the literature. Firstly, specific cutting forces were validated followed by chip morphology.

4.1 Specific cutting forces validation

The average specific cutting forces obtained from FEA simulations were compared with the specific cutting forces obtained from the physical experiments.

M.Asad et al. [50] performed turning experiments on Al2024-T351 workpiece with cutting depth of 4 mm and feed 0.3-0.4 mm/rev for speeds 200 m/min, 400 m/min and 800 m/min with the rake angle 17.5 °, with the clearance angle and tool nose radius to be 7° and 20 µm respectively. The forces are recorded with standard dynamometer and are compared with the FEA results obtained for feed 0.3 and 0.4 mm for cutting speeds 200 m/min, 400 m/min and 800 m/min. The results are shown in Figure 4.1 and the corresponding difference (%) is shown in Figure 4.2.

The difference (%) for the results obtained from M. Asad et al.[50] for uncut chip thickness 0.4mm is ranging from 19.5% to 18.5%, and for uncut chip thickness 0.3 mm the difference observed is ranging from 16.8% to 15.4%. Martin Madaj et al. [51] also conducted experiments for the same cutting parameters used by M. Asad et al. and the forces are shown in Figure 4.1 and the corresponding differences (%) are shown in Figure 4.2. Cutting forces almost remain constant with the increase in cutting speed for the experimental data. Similar trend was observed for the results obtained from finite element simulations.

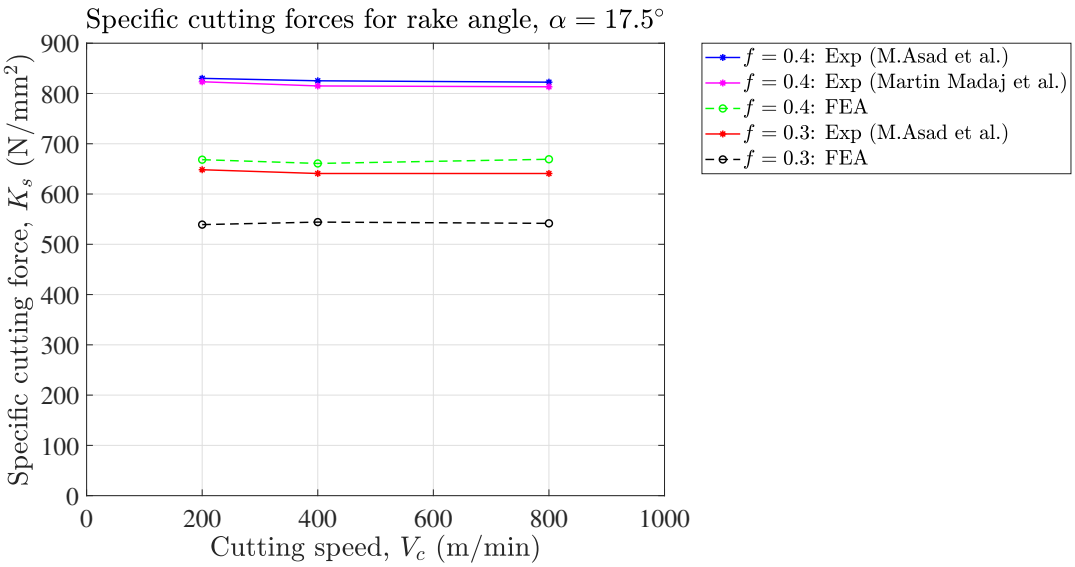


Figure 4.1: Comparison of FEA specific cutting forces with experimental results.

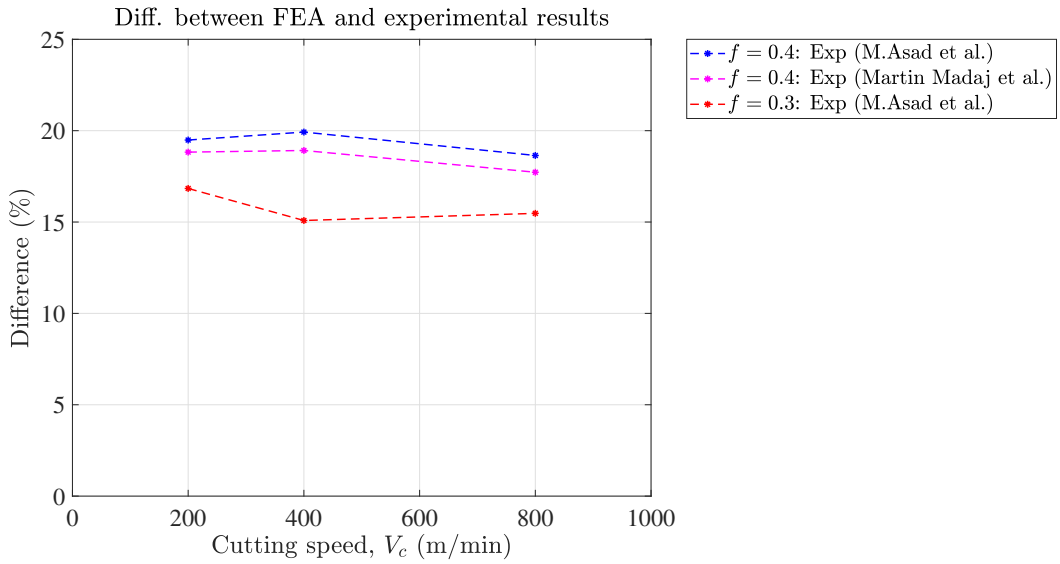
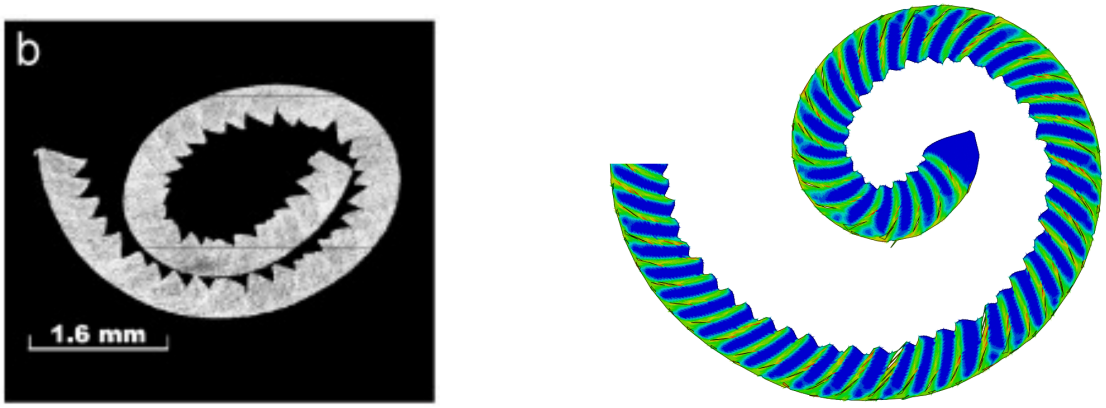


Figure 4.2: Difference in specific cutting forces.

4.2 Chip morphology

Chip morphology is a significant parameter to understand the material behaviour in machining. It can be used as a primary parameter in optimizing the metal cutting process since it reflects the true measure of plastic deformation and provides an estimate of the energy spent in the process [52],[53]. Chip morphology obtained is directly related to the cutting parameters chosen. A high rake angle or a large uncut

chip thickness will result in serrations. Serrations occur due to the instability that arises due to interactions between strain hardening and thermal softening. Figures 4.3a and 4.3b show the chip morphology obtained from the experimental studies [42] and FEA simulations for rake angle 17.5°, uncut chip thickness of 0.4 mm and cutting speed of 800 m/min. The chip obtained from FEA results matches closely with the chip obtained from experimental studies.



(a) Experimental chip [42]

(b) FEA chip

Figure 4.3: Comparison of chip shape obtained from FEA simulations (4.3b) with physical chip obtained from machining experiments (4.3a).

CHAPTER 5: FEA RESULTS AND DISCUSSIONS

5.1 Cutting forces

Cutting forces are important parameters to be determined in machining processes because cutting forces, when multiplied with the displacement of the tool, give external work. The external work is dissipated in the form of plastic work, frictional work, and energy released to form new surfaces during chip formation and chip breakage. Figure 5.1 shows the cutting forces obtained for cutting speed 600 m/min, uncut chip thickness of 0.2mm, and rake angle of 15°. The average cutting force obtained is 493 N. Figure 5.2 shows the cutting forces obtained for cutting speed 400 m/min, uncut chip thickness of 0.3mm, and rake angle of 8°. The average cutting force obtained is 726 N. The cutting forces were calculated by assuming the width of the workpiece to be 4mm.

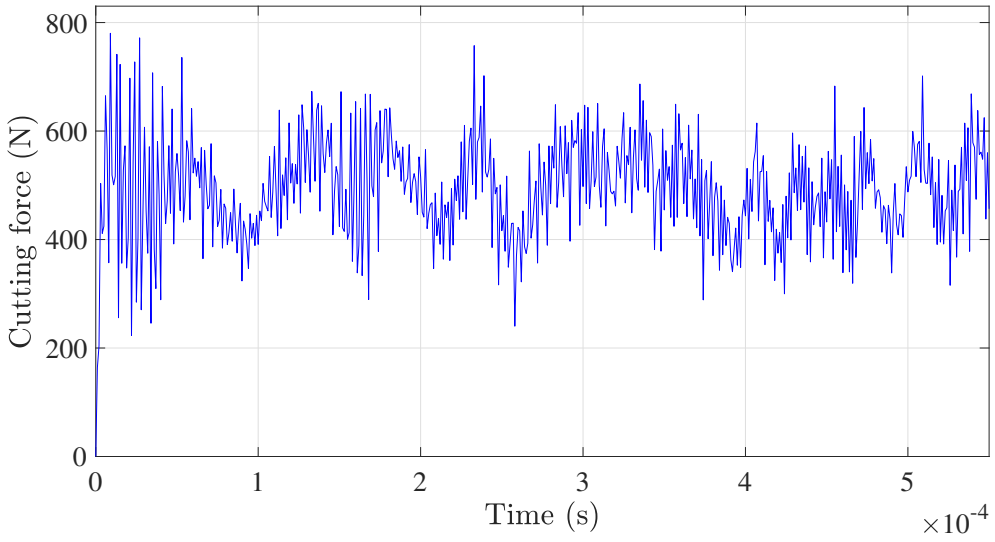


Figure 5.1: Cutting forces for $V_c = 600$ m/min, $\alpha = 15°$ and $f = 0.2$ mm obtained from FEA simulations.

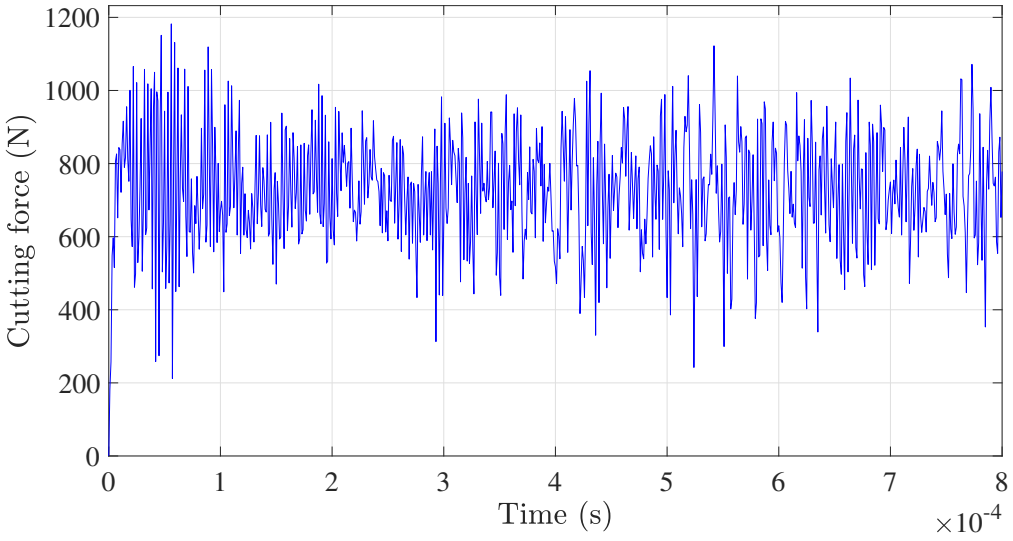


Figure 5.2: Cutting forces for $V_c = 400$ m/min, $\alpha = 8°$ and $f = 0.3$ mm obtained from FEA simulations.

Both the cutting force plots exhibit oscillations(noise), with a deviation of around 20% from the mean. To understand this phenomenon a fast Fourier transform (FFT) plot was necessary. Figure 5.3 represents the magnitude of the FFT plot obtained for the cutting forces shown in Figure 5.2. The amplitude at frequency 0Hz (highest peak) corresponds to the mean cutting force, 726N, whereas the amplitudes at high frequencies corresponds to the noise.

A filter was applied to this FFT plot to remove high frequencies, with the cutoff frequency of 0.05 MHz, which is shown in Figure 5.4. Later, inverse FFT was applied to the filtered FFT plot to compare the actual cutting forces with the filtered cutting forces. Figure 5.5 shows the actual and filtered cutting forces, the filtered cutting force exhibits less noise than the actual cutting force and the difference between their mean cutting forces is 0.01N. Hence, to reduce the noise, the high frequencies should be filtered.

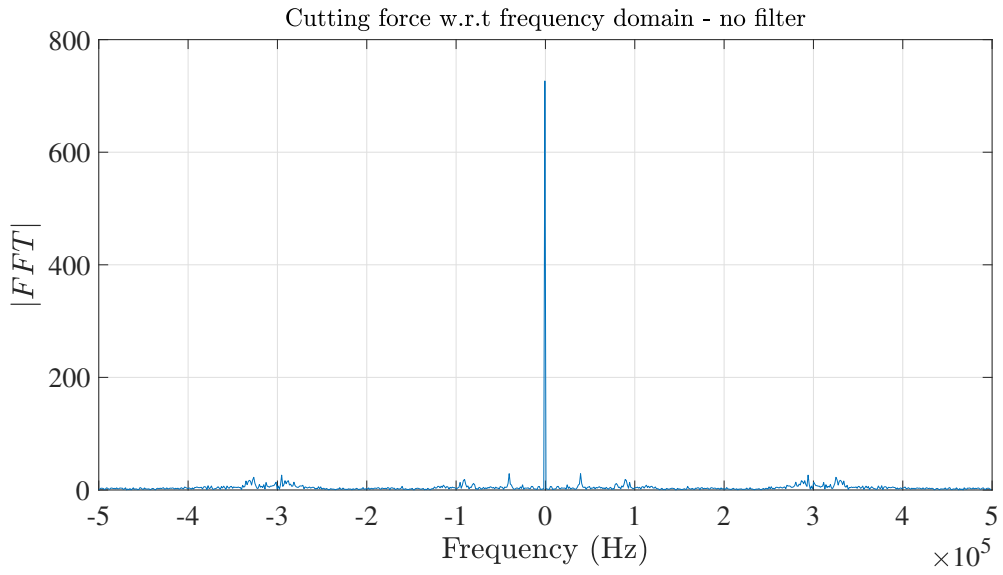


Figure 5.3: Fast Fourier transform for the cutting forces shown in Figure 5.2.

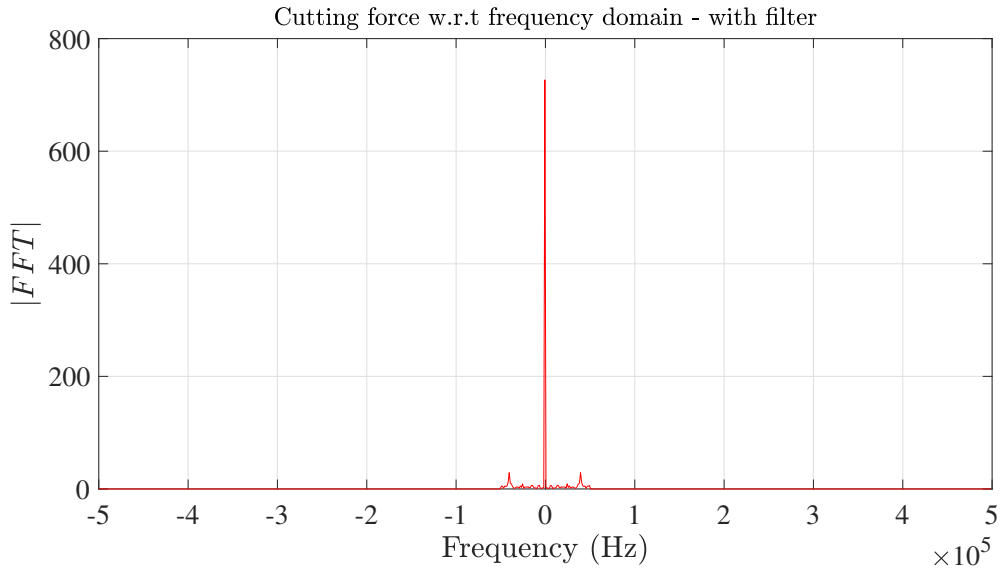


Figure 5.4: FFT obtained after applying filter for the FFT shown in Figure 5.3.

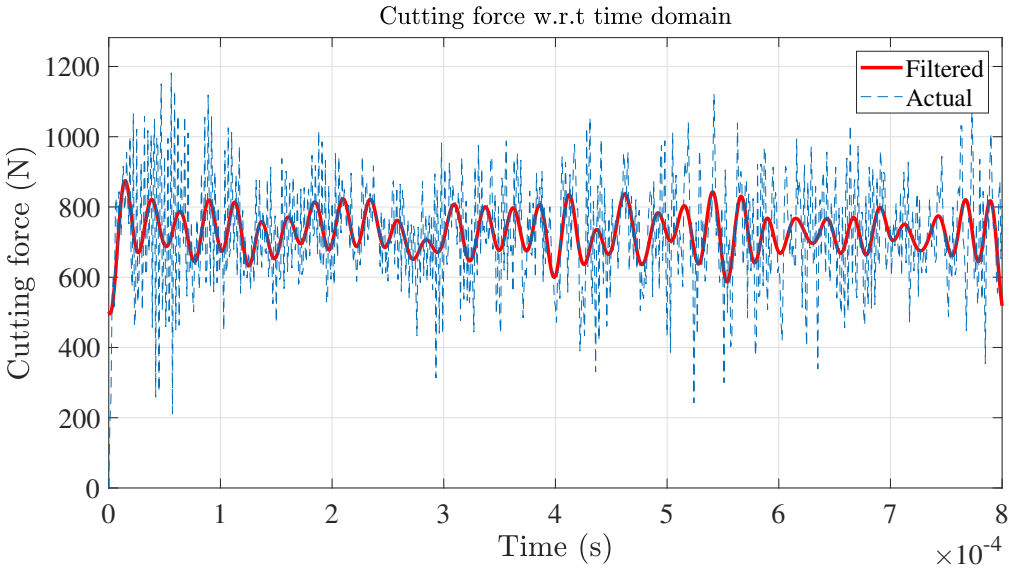


Figure 5.5: Actual cutting force and filtered cutting force.

5.2 Specific cutting forces

Specific cutting forces (K_s) play a significant role in understanding the machining process. They are obtained by dividing the average cutting force by the product of uncut chip thickness and width of the workpiece.

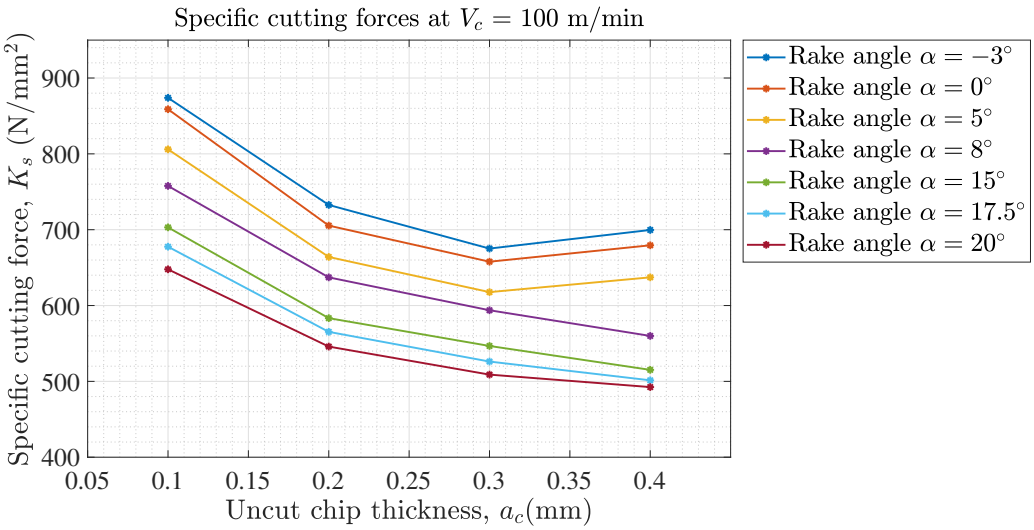


Figure 5.6: Variation of specific cutting forces with uncut chip thickness and rake angles at $V_c = 100$ m/min.

Figure 5.6 represents the variation of specific cutting forces with uncut chip thick-

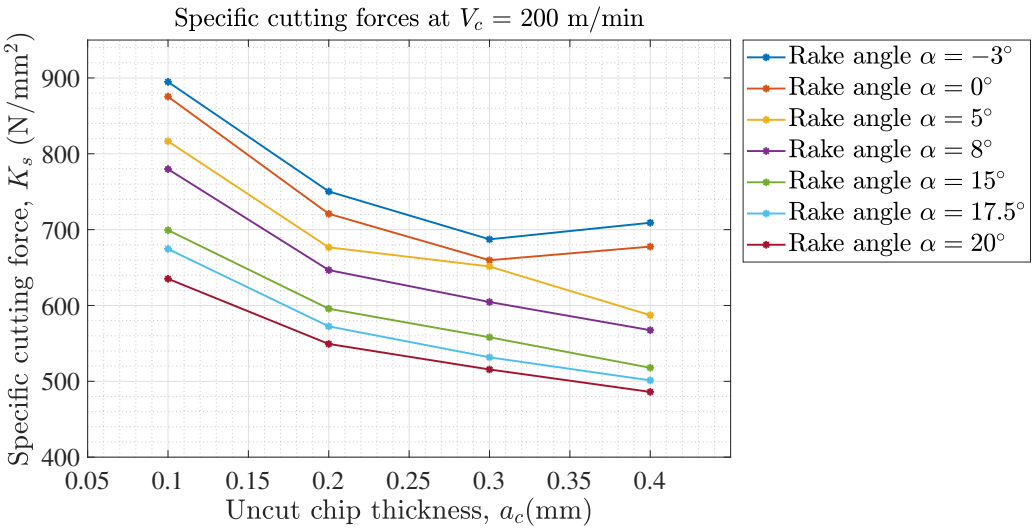


Figure 5.7: Variation of specific cutting forces with uncut chip thickness and rake angles at $V_c = 200$ m/min.

ness for various rake angles during the cutting speed of 100 m/min. It is observed that specific cutting forces decrease with the increase in uncut chip thickness and the increase in rake angles.

A similar observation can be made from Figure 5.7, for the cutting speed of 200 m/min, where the specific cutting forces decrease with the increase in uncut chip thickness and rake angle. Similar observations were made by Marusich in [54] and by Parle in [55]. With respect to Figures 5.6 and 5.7 the highest K_s was obtained for rake angle -3° and uncut chip thickness 0.1 mm, whereas the least K_s was obtained for rake angle of 20° and uncut chip thickness of 0.4 mm for Figure 5.7.

Figures 5.8, 5.9, 5.10 and 5.11 show the variation of specific cutting forces with respect to the cutting speeds at various rake angles.

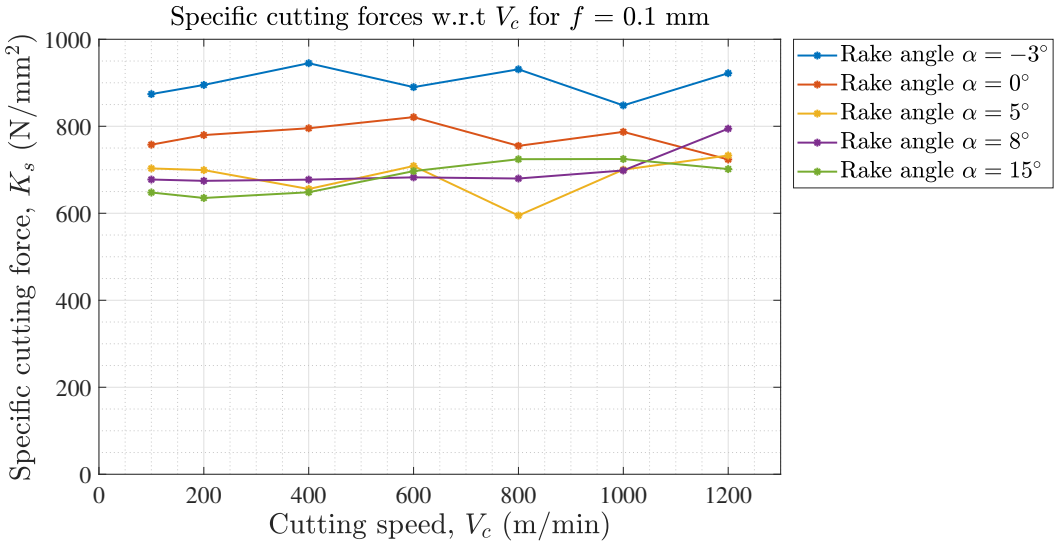


Figure 5.8: Variation of specific cutting forces with cutting speeds and rake angles at $f = 0.1$ mm.

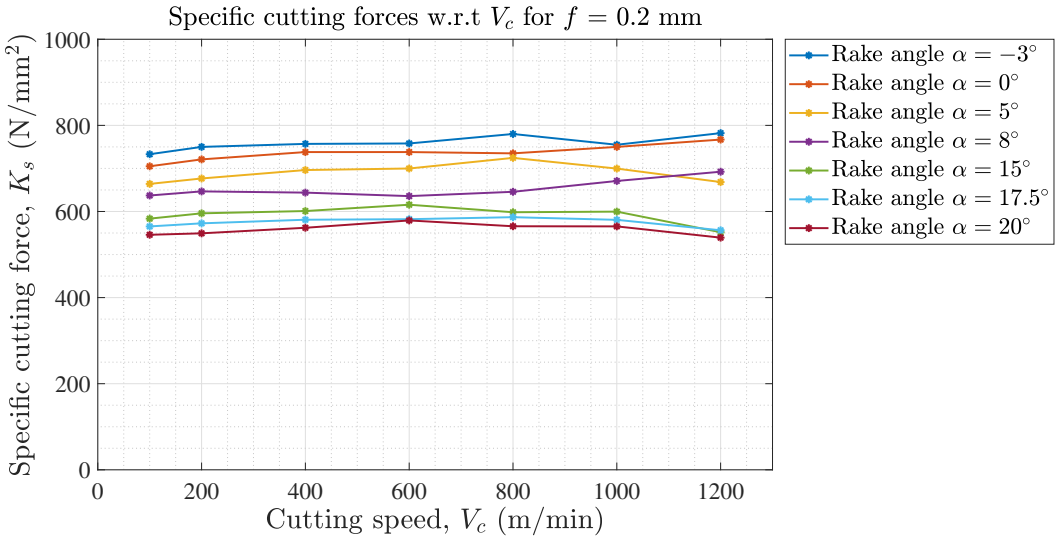


Figure 5.9: Variation of specific cutting forces with cutting speeds and rake angles at $f = 0.2$ mm.

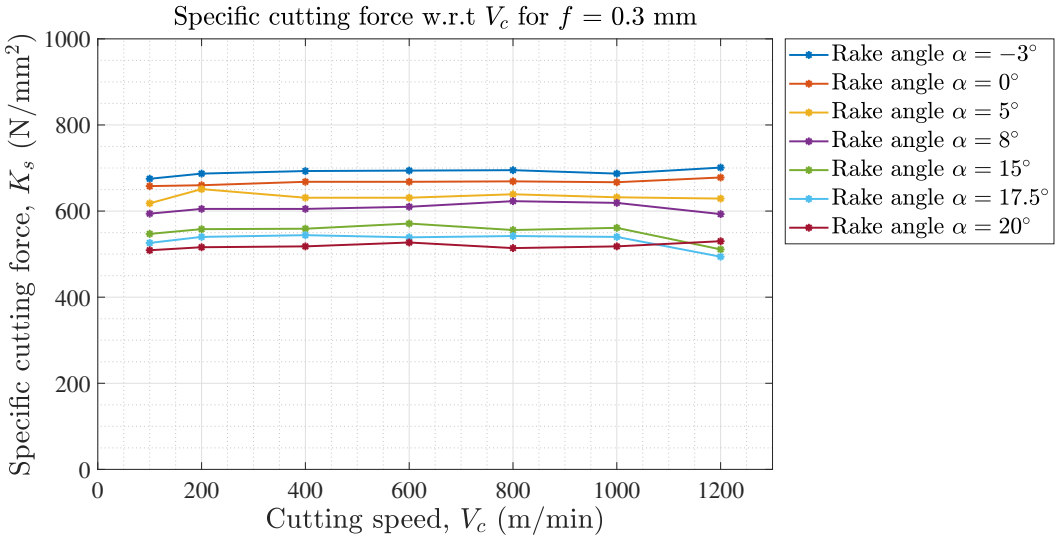


Figure 5.10: Variation of specific cutting forces with cutting speeds and rake angles at $f = 0.3$ mm.

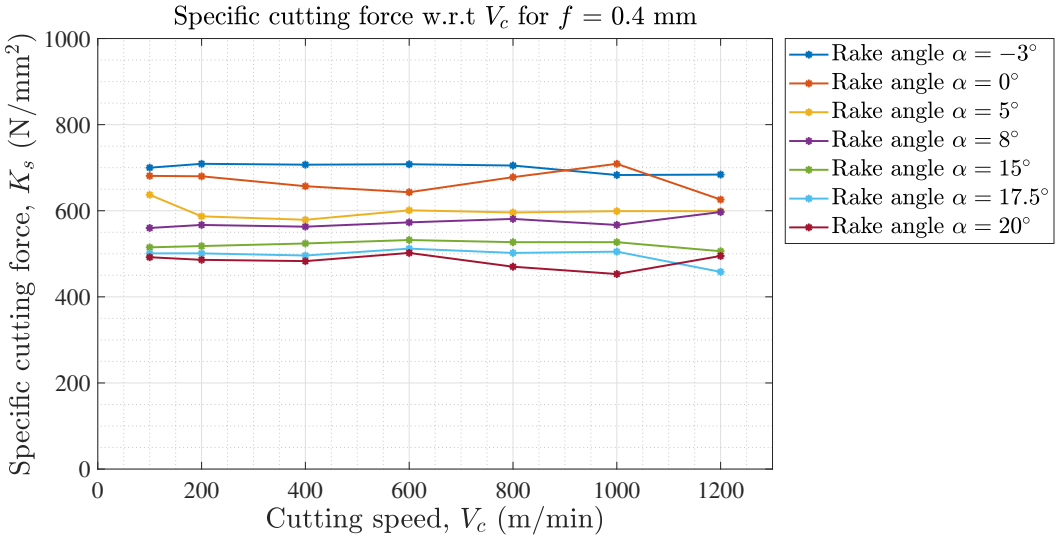


Figure 5.11: Variation of specific cutting forces with cutting speeds and rake angles at $f = 0.4$ mm.

From all the above plots (K_s variation with V_c), it can be inferred that the cutting speeds have a moderate influence on forces. Similar observations were made by Lucca et al. in [56], M.Asad et al in [50] and Martin Madaj et al. in [51]. It is important to note that the average cutting force used in the computation of specific cutting forces were obtained from the FEA-generated force data. It may be recalled that this force

data contains significant noise with high amplitudes and high frequencies (see Figures 5.1 and 5.2). The average cutting force is independent of the high frequencies as the FFT computations, discussed previously in Figures 5.3, 5.4, and 5.5, show. The error bars on the specific cutting force values can be calculated from the cutting force data after filtering out the high frequency components.

5.3 Maximum tool temperatures

In metal machining, the rise in temperature is mainly contributed by the heat generated due to plastic work in primary shear zone and friction at tool-chip interface. Temperature results mainly depend on the cutting parameters used for machining along with the thermal properties of the tool and workpiece. Maximum amount of heat is taken by the chip, 10% to 20% goes to the tool and some heat is taken by the shank. Cutting temperatures can be controlled by proper selection of material and geometry of the cutting tool. Tool chip interface has the highest temperature, and its value is maximum almost at the middle of the chip-tool contact length.

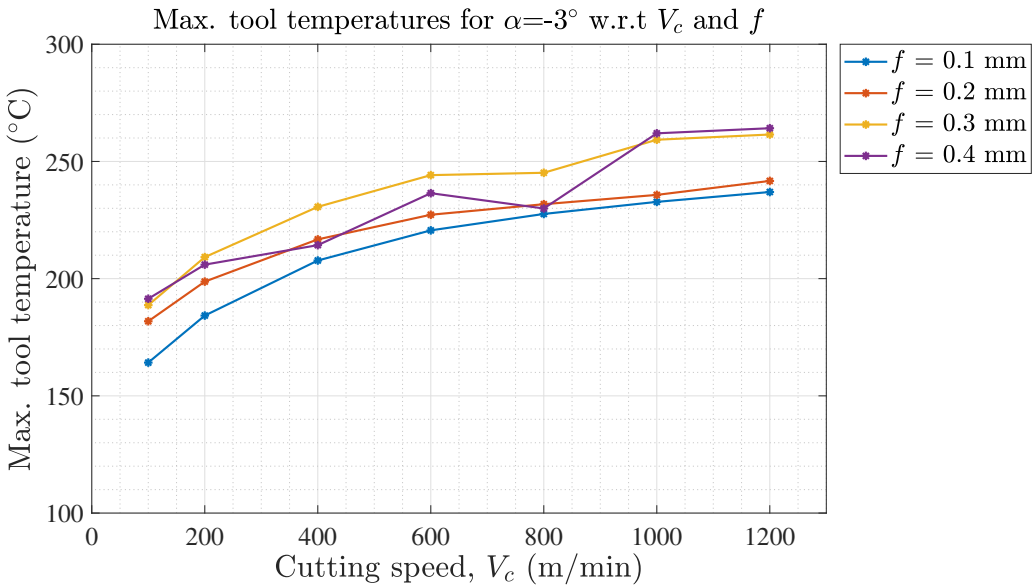


Figure 5.12: Variation of maximum tool temperatures with cutting speeds and uncut chip thickness at $\alpha = -3°$.

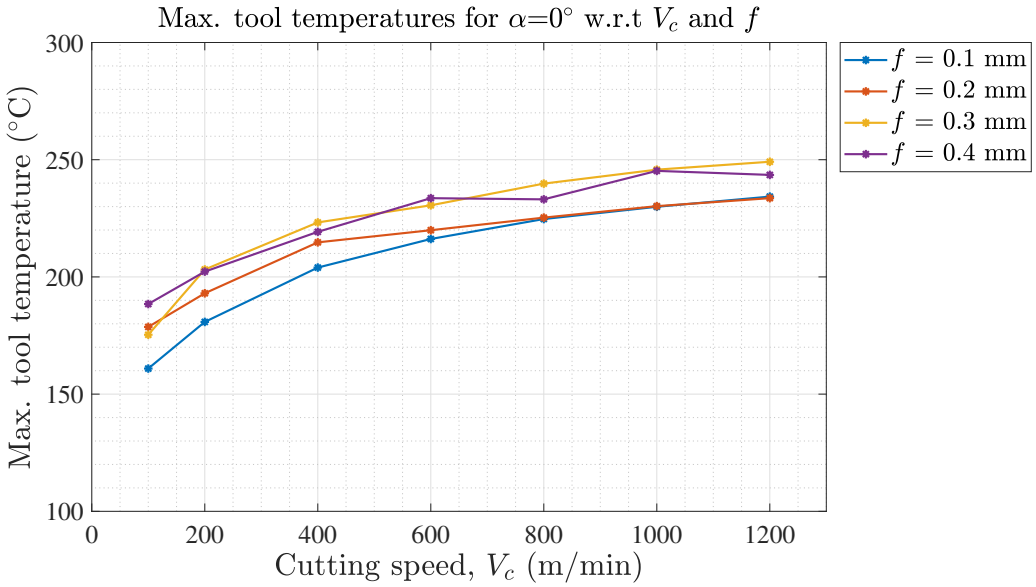


Figure 5.13: Variation of maximum tool temperatures with cutting speeds and uncut chip thickness at $\alpha = 0°$.

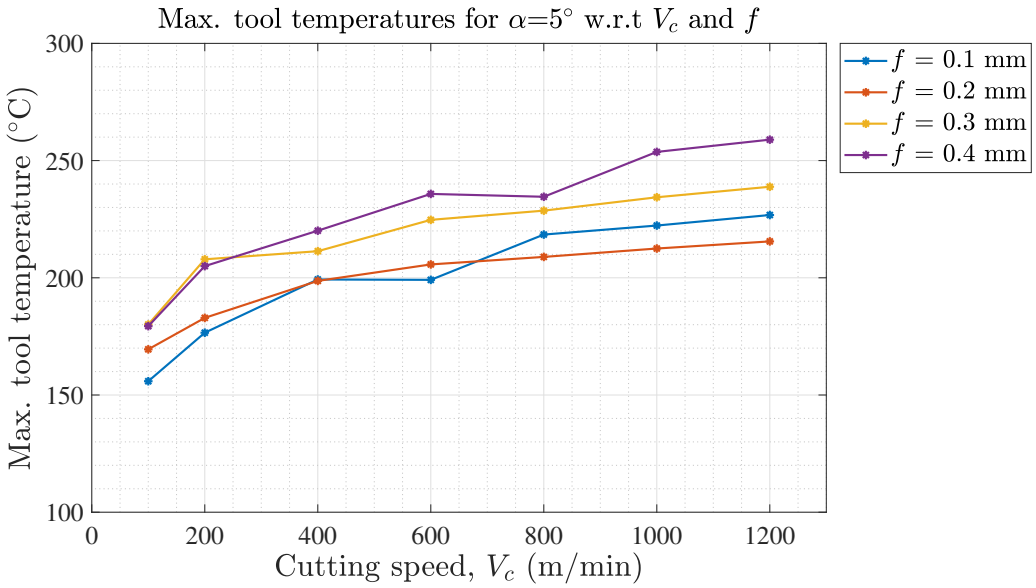


Figure 5.14: Variation of maximum tool temperatures with cutting speeds and uncut chip thickness at $\alpha = 5°$.

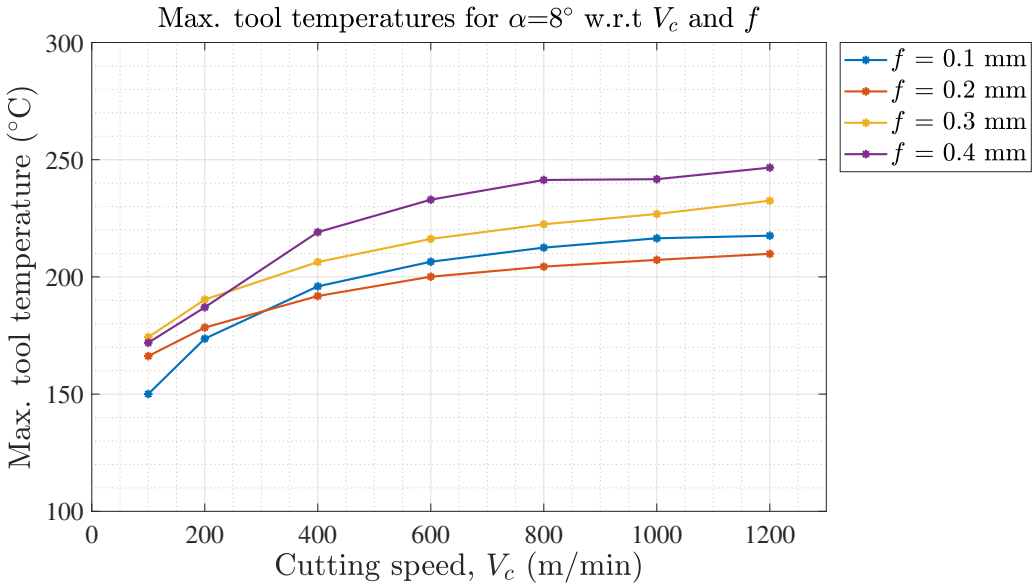


Figure 5.15: Variation of maximum tool temperatures with cutting speeds and uncut chip thickness at $\alpha = 8°$.

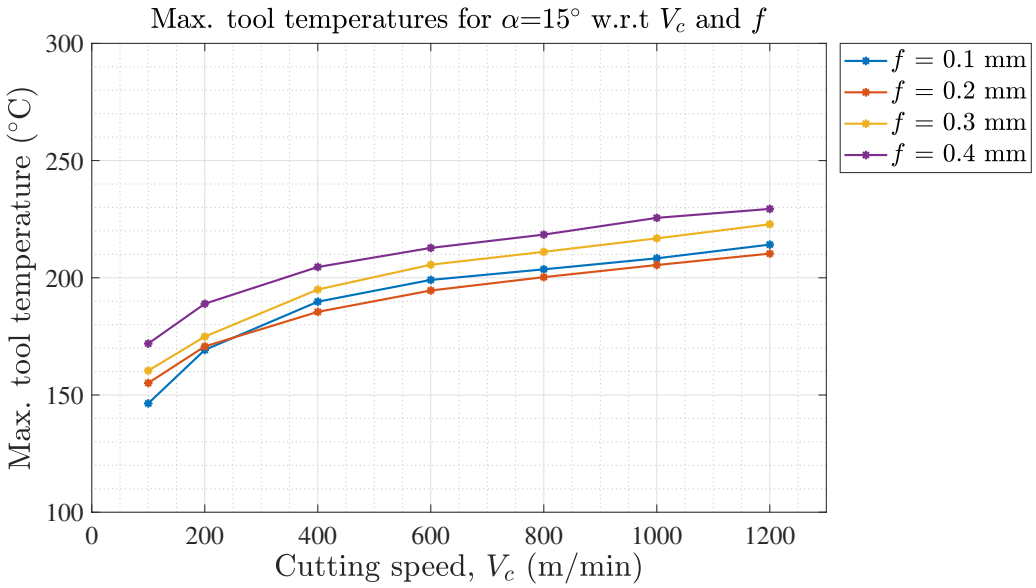


Figure 5.16: Variation of maximum tool temperatures with cutting speeds and uncut chip thickness at $\alpha = 15°$.

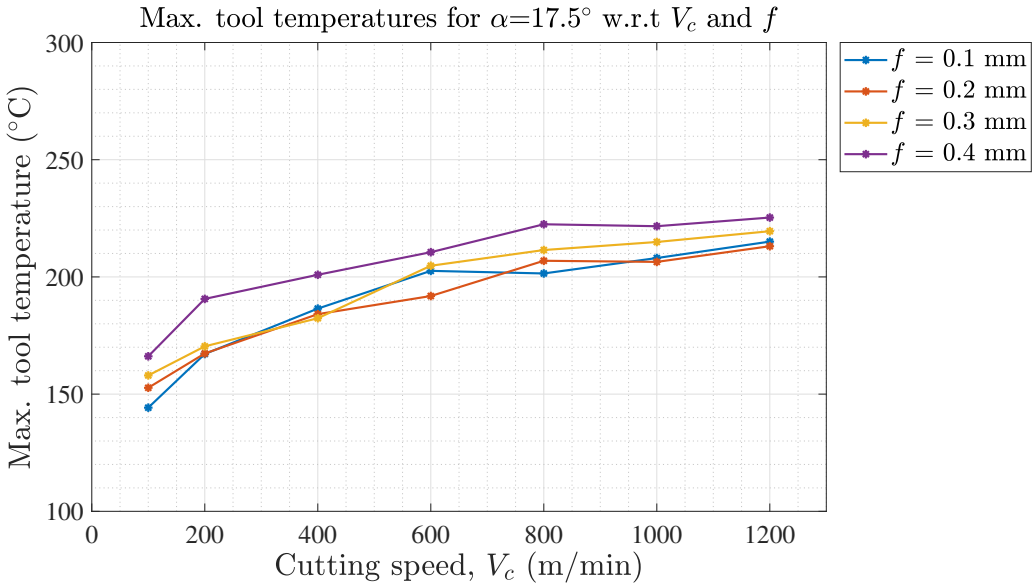


Figure 5.17: Variation of maximum tool temperatures with cutting speeds and uncut chip thickness at $\alpha = 17.5°$.

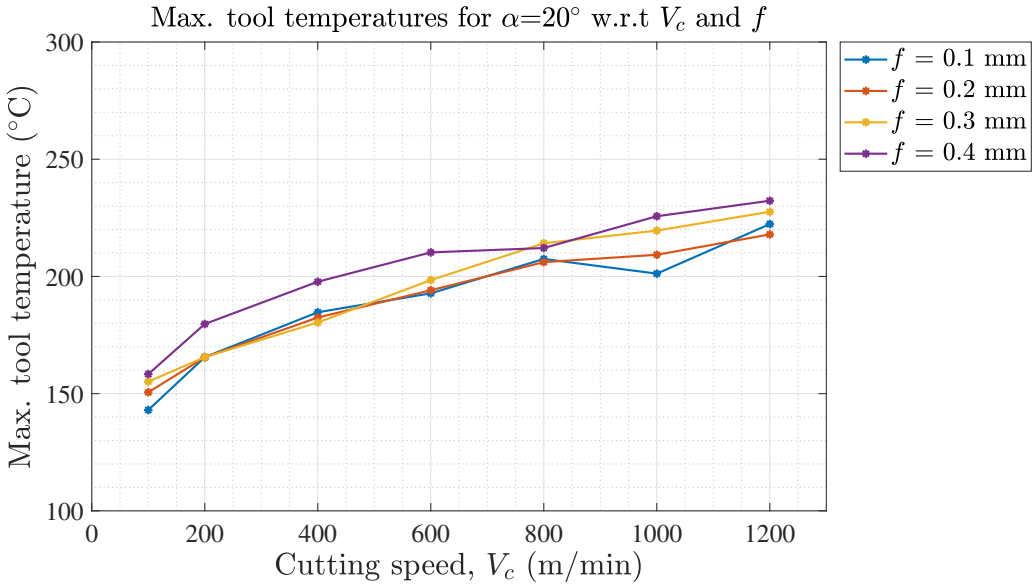
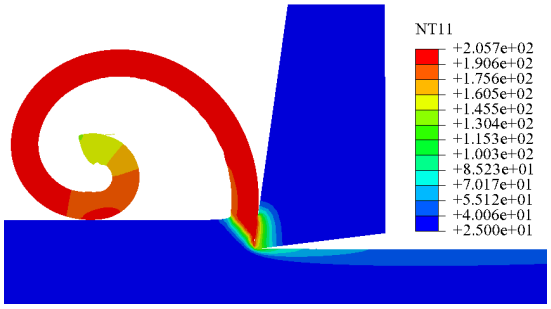


Figure 5.18: Variation of maximum tool temperatures with cutting speeds and uncut chip thickness at $\alpha = 20°$.

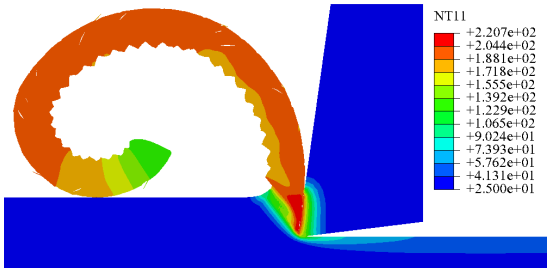
In this work, average maximum tool temperatures were recorded for various rake angles, uncut chip thickness, and cutting speeds. Figure 5.12 represents the variation in maximum tool temperature for rake angle $-3°$ at various cutting speeds and uncut

chip thickness. It is observed that temperatures increase with the increase in cutting speeds and uncut chip thickness. A similar trend was seen for other rake angles, as shown in Figures 5.13, 5.14, 5.15, 5.16, 5.17 and 5.18, this trend is due to the increase in friction with the increase in uncut chip thickness. Out of all the simulations the maximum tool temperature of 264 °C was obtained for rake angle -3° and 0.4 mm uncut chip thickness during the cutting speed of 1200 m/min.

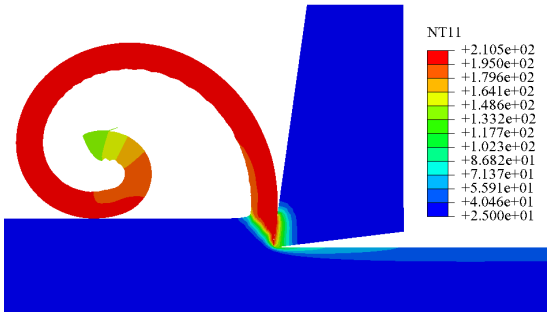
Figure 5.19 shows temperature profiles obtained from FEA simulations. As uncut chip thickness increases at constant speed, the temperature values also increase, see figure pairs (5.19a & 5.19b), (5.19c & 5.19d), and (5.19e & 5.19f)). At the same time, as cutting speed increases with constant uncut chip thickness, temperatures also increase, see Figures (5.19a, 5.19c & 5.19e) and (5.19b, 5.19d & 5.19f).



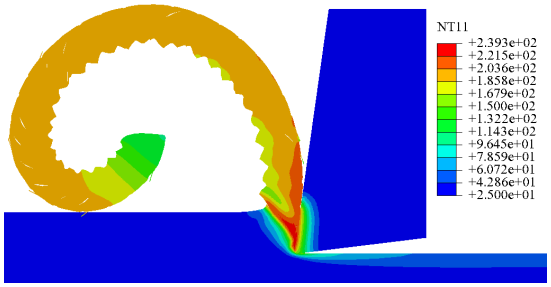
(a) $V_c = 400$ m/min, $f = 0.2$ mm.



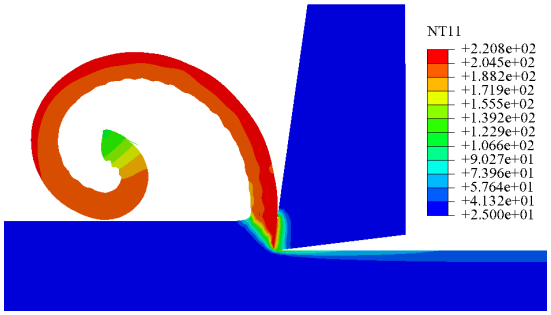
(b) $V_c = 400$ m/min, $f = 0.3$ mm.



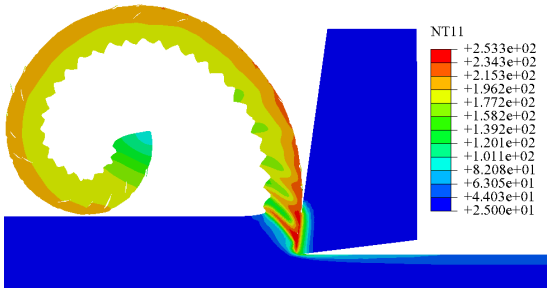
(c) $V_c = 600$ m/min, $f = 0.2$ mm.



(d) $V_c = 600$ m/min, $f = 0.3$ mm.



(e) $V_c = 1200$ m/min, $f = 0.2$ mm.



(f) $V_c = 1200$ m/min, $f = 0.3$ mm.

Figure 5.19: Temperature profiles obtained from FEA simulations for rake angle $\alpha = 8°$, uncut chip thickness $f = 0.2$ mm and 0.3 mm at cutting speeds $V_c = 400$ m/min, 600 m/min and 1200 m/min respectively.

CHAPTER 6: ARTIFICIAL NEURAL NETWORK MODELING

This chapter presents in detail the parameters, features, and phases that go into building a neural network. In this work, we predicted specific cutting forces and maximum tool temperatures with the input parameters rake angle, uncut chip thickness, and cutting speed. Two different neural networks were constructed to predict both specific cutting forces and maximum tool temperatures. The first one is shallow and wide, and the second one is deep and narrow. Both the networks are based on feed-forward backpropagation mechanism. Five activation functions were used during the training of the neural networks. The trained networks were tested by using the test data set and the network that produced the least error was identified.

6.1 Artificial neural networks (ANN)

The ANN approach followed here for predicting specific cutting forces and tool temperatures implements a supervised learning model, where the model learns by using input-output pairs in the training phase. The supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new data. When the output is used to create categories or classes, the problem is called a classification problem. When the output is real, continuous value, it is a regression problem. As both the specific cutting forces and maximum tool temperatures are numerical and continuous values, a regression problem is solved.

An artificial neural network structure consists of three main layers: input, output and hidden layers, as shown in Figure 6.1. The first (left) layer is the input layer, and the last (right) layer is the output layer. The layer in between is the hidden layer. If the number of hidden layers are more than one, then the network is said to be a deep

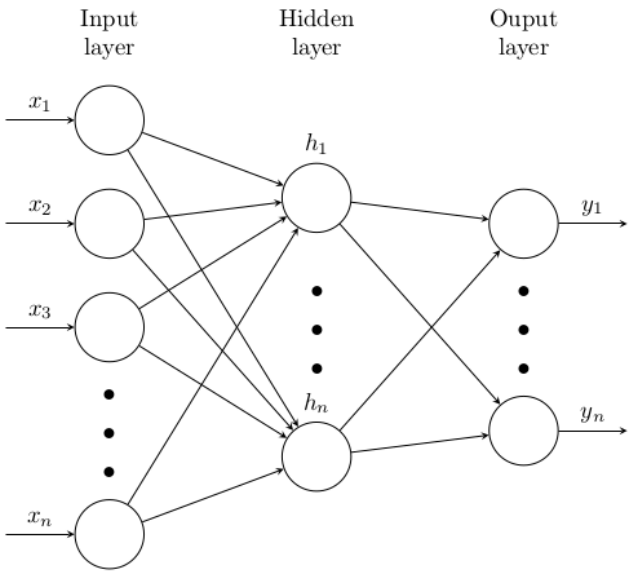


Figure 6.1: An example of an artificial neural network (ANN).

neural network (DNN). If the network has only one hidden layer, as shown in Figure 6.1, it is called a shallow neural network (SNN).

Each of these layers have components called neurons (the ones shown as a circle in the Figure 6.1). The neurons present in a layer are linked to the neurons in the immediate preceding and succeeding layer through connections, which are called weights or synapses. Connections only exist between neurons of two adjacent layers, and there are no connections present between neurons in the same layer.

6.1.1 ANN mechanism

Figure 6.2 represents an ANN with 4 neurons in the input layer, 2 neurons in the hidden layer, and an output layer with 1 neuron.

The user has to define the neurons present in the first layer. The values for neurons available in the following layer are obtained by taking the values in the first layer and computing their weighted sum and adding a constant (b), referred as bias, denoted by Z_1. Then, an activation function, g, is applied on the whole expression (shown as $g(Z_1)$) to bring it down into a certain range. The same mechanism is repeated for the other neuron present in the layer.

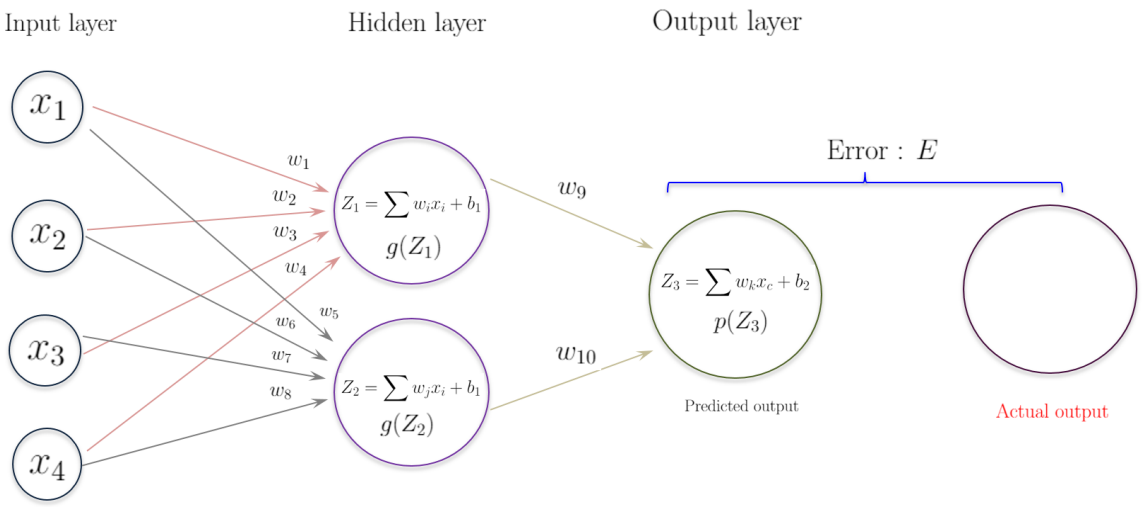


Figure 6.2: An example of ANN with 4 neurons in the input layer, 2 neurons in the hidden layer and 1 neuron in the output layer.

Now, both the neurons present in the hidden layer serve as inputs for the neurons present in the next layer. The same mechanism is repeated until the final output layer is reached. The difference between the actual output and the predicted output is the error, and it is calculated using a cost (or error) function. Since the predicted values depend upon the weights and biases, it is clear that the error function, E, is also a function of the weights and biases for a given set of training data, ie., $E = E(w, b)$. By absorbing the bias, b, into the weights as an additional parameter, E can be assumed to be a function of only weights w_i. If the error is not acceptable, the weights are updated through various methods, depending upon the learning algorithm used.

One of the approaches used widely is the gradient descent method, where the weight updates are computed using the derivatives of the error function with respect to the weights:

$$w_i^{(j+1)} = w_i^{(j+1)} - \eta \frac{\partial E}{\partial w_i} \Bigg|^{(j)} \tag{6.1}$$

In the Equation 6.1, η is the learning rate that is used to control the magnitudes of the corrections applied to w_i and j is the iteration number. If the value of η is too large the model is prone to convergence issues. At the same time, if the value is too

small the computational time and cost increases. The updated weights are used to calculate the errors in prediction. The process is repeated until the error is less than a pre-selected value or a maximum number of iterations has been reached. Although Equation 6.1 captures the essence of weight updates, in a typical ANN with multiple hidden layers, the gradient calculation is quite complicated and involved.

6.2 Programming language used

Python programming language was used for building the neural networks in this work. Python is widely used for many engineering applications around the world as it is a powerful, simple, open source programming language. The time spent on writing and debugging python is considerably less compared to other programming languages, such as C, C++ and Java. Python comes with a great amount of inbuilt libraries, which makes the language user friendly. Another important feature of python is its capacity to interact with third party languages and platforms. Additionally, there is a big community and ecosystem around Python, which results in a diverse set of available tools oriented to different kind of tasks.

Keras is the library that was used in this work to build the neural networks. It is a high-level neural networks Application programming interface (API), written in Python. Keras supports both central processing unit (CPU) and Graphics Processing Unit (GPU) computations [57]. Keras allows for easy and fast prototyping through user friendliness, modularity, and extensibility [58]. The python code is written using google colab, a free cloud based service based on Jupyter notebook environment that supports free GPU. It is a valuable tool for experimenting with neural networks. All the libraries are pre-installed. The Tables 6.1 and 6.2 provide more information regarding the software and the list of libraries that were used in work for building neural networks.

Table 6.1: Used software's and there versions.

Coding language	**Version**
Python	3.6

Table 6.2: List of libraries used.

Name	**Purpose**
Pandas	Loading data & saving results
Scikit-learn	Data preparation and analysis
Keras	Building the neural networks

6.3 Data for building neural network

In this work, the data for building neural networks was not obtained from physical experiments but was obtained from finite element simulations. The finite element simulations were performed by changing rake angles, uncut chip thickness and cutting speeds. The detailed explanation about simulations and data extraction is presented in the section 3.7 of the chapter 3. The data shown in the tables 3.5, 3.6, 3.7 and 3.8 was normalized according to the Equation 6.2. Normalization is implemented to ensure data sets are present in a logical correlation. If they are not normalized, the network could possibly consider the data set with higher arithmetic value to be more significant than others. This may affect the generalization ability of the network and can also lead to over fitting [9]. Later the normalized data is split into training and testing using a 80:20 ratio. A further 10% of validation split was performed on the training data set. This was determined to be the most reasonable split after trying different proportions. Thus, the training set consisted of 140 sets, while the validation and test sets consisted of 16 and 40 sets, respectively. During the prediction of specific cutting forces, the maximum tool temperature column from the tables 3.5, 3.6, 3.7, 3.8 was excluded, and vice versa, during the prediction of maximum tool temperature.

$$V_{\mathrm{N}} = \frac{V - V_{\mathrm{min}}}{V_{\mathrm{max}} - V_{\mathrm{min}}} \tag{6.2}$$

6.4 Adam optimizer

An optimizer or optimization algorithm is used to minimize the error function by updating the weights and biases after every iteration until the cost function reaches a global optimum or until the completion of specified number of epochs. Several learning algorithms are available in the literature. Recently, Adaptive Moment Estimation (Adam) is one such learning algorithm that has been used extensively.

Adam is an algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. Adam computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. The algorithm is straight forward to implement and has little memory requirements [59]. Adam is the combination of two other stochastic gradient descent algorithms [59] shown below:

- Adaptive gradient algorithm (AdaGrad): This algorithm improves performance on problems having sparse gradients because it maintains a per-parameter learning rate.
- Root mean square propagation (RMSProp): This algorithm has a similar approach followed by AdaGrad with a minor change as the per-parameter learning rates are adapted based on the average of recent magnitudes of the gradients for the weight.

Instead of adapting the parameter learning rates based on the average first moment (the mean) as in RMSProp, Adam makes use of the average of the second moments of the gradients (the uncentered variance). The algorithm, calculates an exponential moving average of the gradient and the squared gradient, and the parameters β_1 and β_2 control the decay rates of these moving averages. The configuration parameters are:

- (l_r): Referred as the learning rate or step size.

- (β_1): The exponential decay rate for the first moment estimates (e.g. 0.9)
- (β_2): The exponential decay rate for the second-moment estimates (e.g. 0.999). This value should be close to 1.
- (ϵ): A very small number to prevent any division by zero in the implementation (e.g. 10^{-8})
- decay: Learning rate decay over each update.

In this work, Adam learning algorithm was implemented by the library Keras and the default parameters used are shown in the Table 6.3.

Table 6.3: Parameters for implementing Adam optimizer in Keras.

l_r	β_1	β_2	ϵ	decay
0.001	0.9	0.999	10^{-8}	0.0

6.5 Epochs and batch size

Before feeding the data sets to the neural networks the number of epochs and batch size have to be specified. Before defining epochs and batch size it is crucial to understand the sample. A sample is a single row of data, and it contains inputs that are given to the input layer and an output. Regarding batch size, it is a hyperparameter that defines the number of samples that are fed to the network. It can be thought of as a for-loop iterating over one or more samples and making predictions. At the end of each batch size, the predictions are compared to the actual outputs and errors are computed. The errors computed are used by the optimizer/learning algorithm to update the weights. A training dataset can be divided into many batches.

Epoch is another hyperparameter that defines the total number of times the learning algorithm/optimizer is passed to the network. In other words, it can be defined as one forward pass and one backward pass of all the training examples. An epoch is comprised of one or more batches. In this work, the number of batch size and the

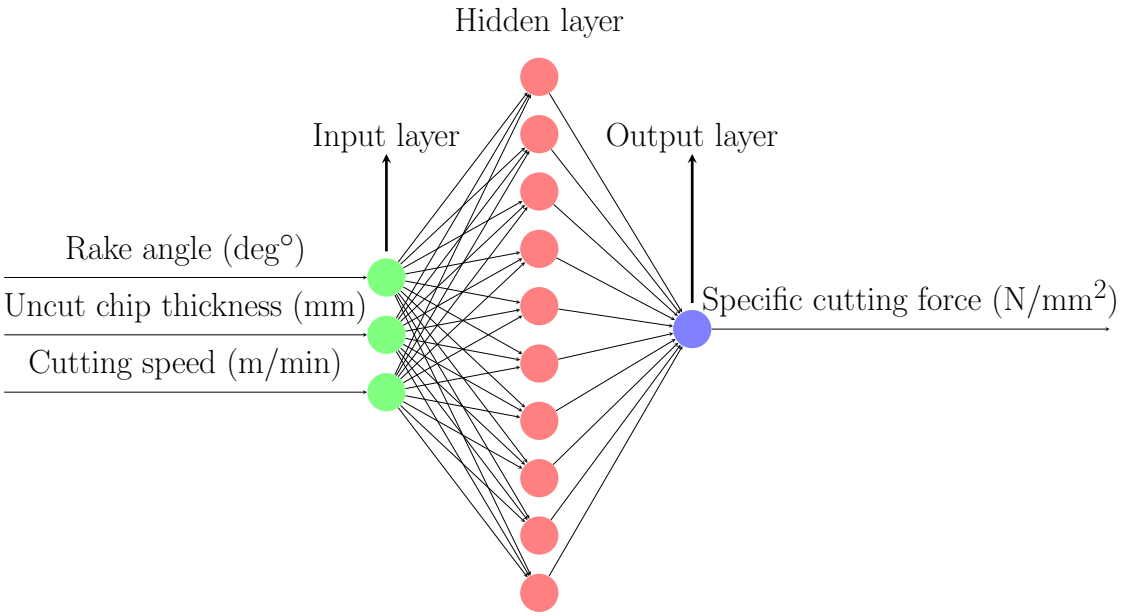
epochs are determined by trial and error approach. The epochs are determined to be 150, whereas the batch size is 20.

6.6 Neural networks in this work

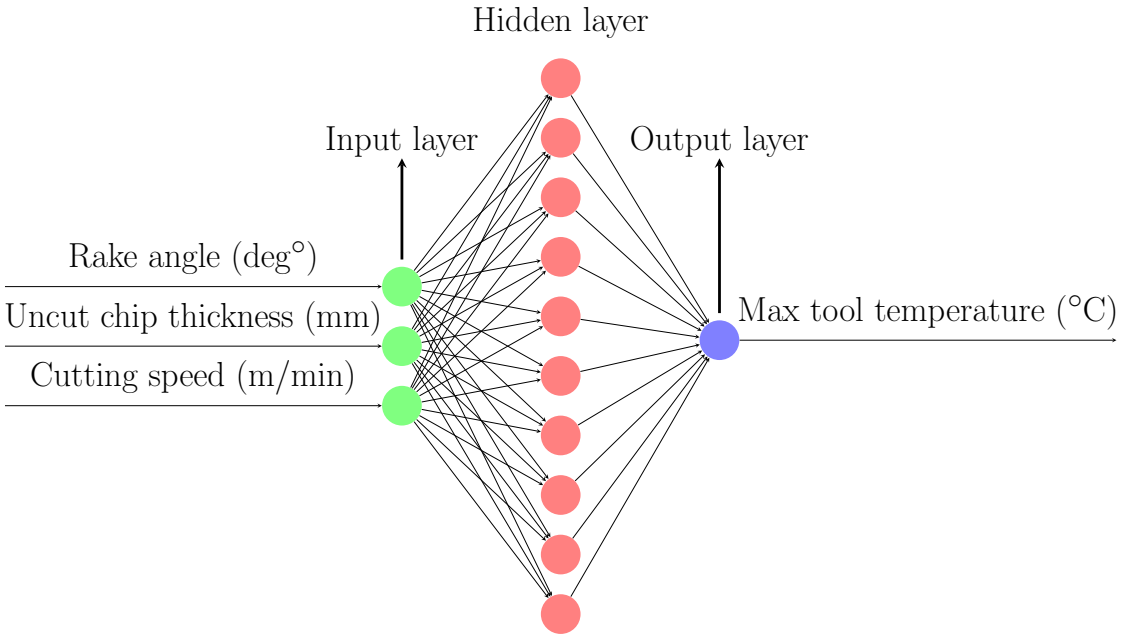
Two types of neural networks were built and tested in this work. They are:

- Shallow neural networks (SNN) : In these networks there is only single hidden layer present. An example of this structure is shown in the Figure 6.3. The number of neurons present in the hidden layer can be increased or decreased. In short, SNNs have an input layer, one hidden layer and an output layer.

- Deep neural networks (DNN) : In these networks there are at least two hidden layers, and the neurons present in each of these hidden layers can be varied. Additional hidden layers can be added to the network depending upon the complexity of the data available. An example of this network is shown in the Figure 6.4.

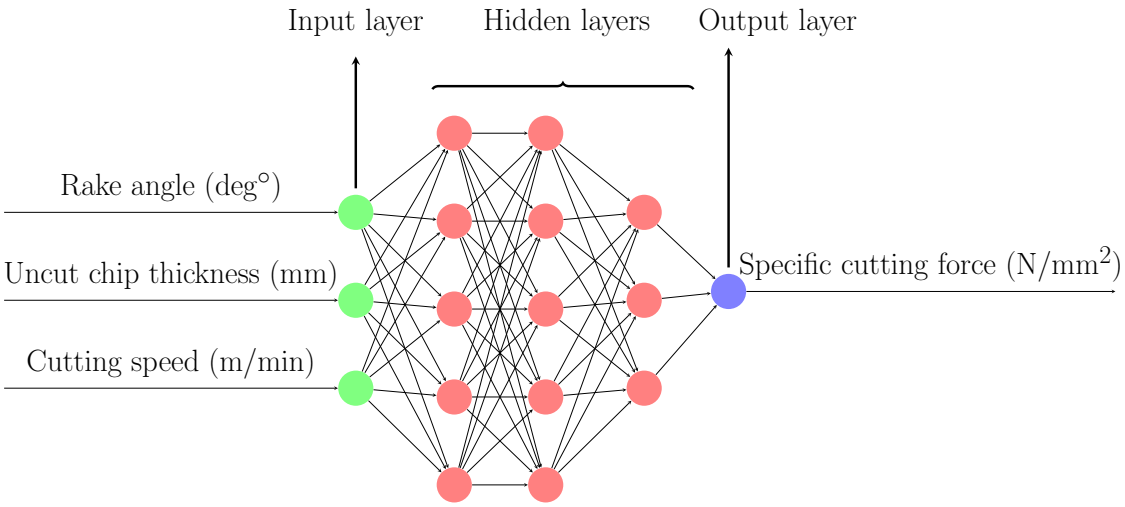


(a) Example of a shallow neural network predicting specific cutting force.

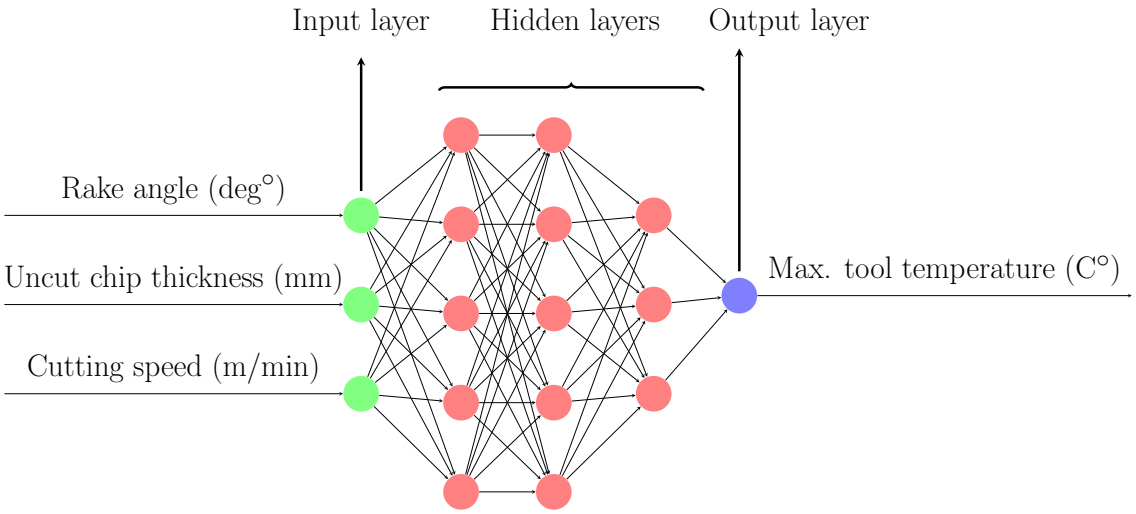


(b) Example of a shallow neural network predicting maximum tool temperature.

Figure 6.3: Shallow neural networks.



(a) Example of a deep neural network predicting specific cutting force.



(b) Example of a deep neural network predicting maximum tool temperature.

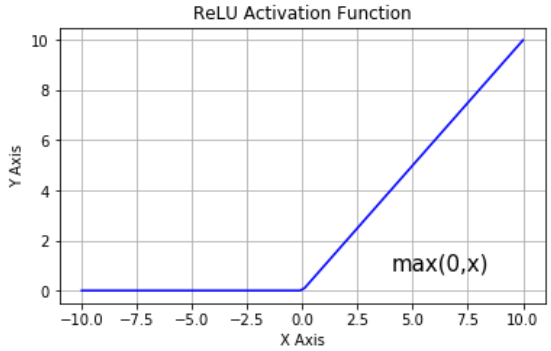
Figure 6.4: Deep neural networks.

6.7 Activation functions

Activation functions, also known as transfer functions, are crucial components in building an ANN. Activation function is a curve that is used to map the values of the network between bounded values. This is applied for every neuron in the network. Typically, activation function has a squashing effect. There are many activation functions that are available in various neural network packages. The following activation

functions were implemented while building the neural networks.

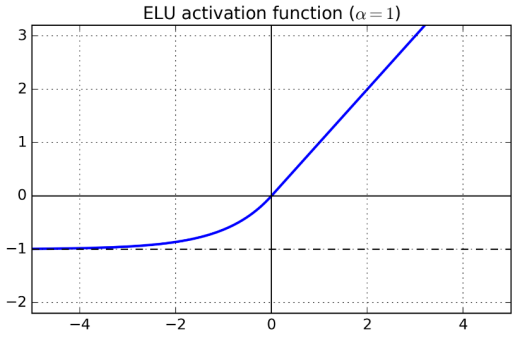
- Rectified Linear Units (ReLU): Figure 6.5 shows the plot of this activation function along with its equation and range.



Equation	Range
$f(x) = max(0, x)$	$[0, \infty)$

Figure 6.5: ReLU - plot, equation & range.

- Exponential Linear Unit (ELU): Figure 6.6 represents the activation function along with its equation and range.



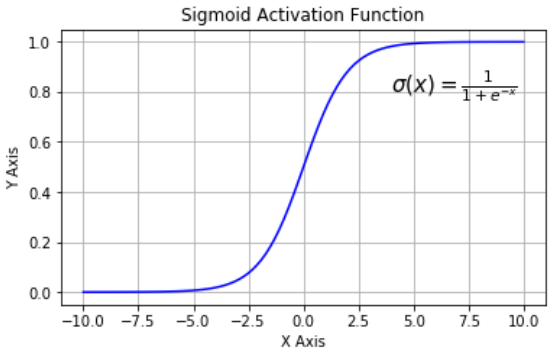
Equation	Range
$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$(-\alpha, \infty)$

Figure 6.6: ELU - plot, equation & range.

- Sigmoid : Figure 6.7 represents the activation function along with its equation and range.

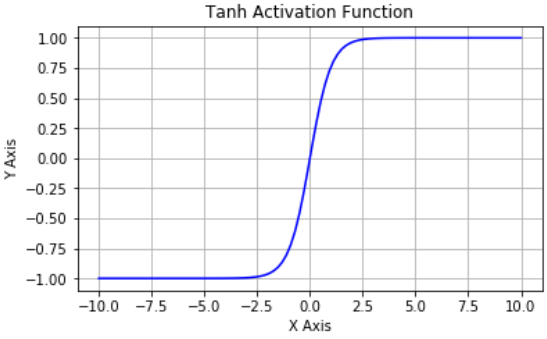
- Hyperbolic tangent (Tanh) : Figure 6.8 represents the activation function along with its equation and range.

- Linear : Figure 6.9 represents the activation function along with its equation and range.



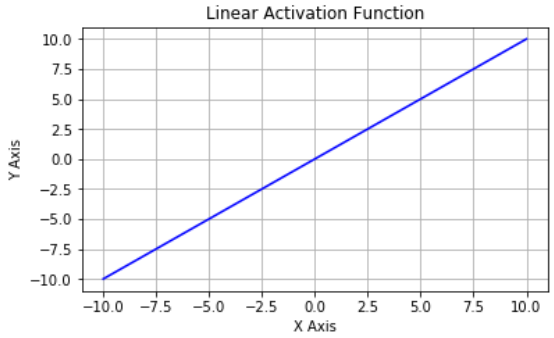
Equation	Range
$f(x) = \dfrac{1}{(1+e^{-x})}$	$(0, 1)$

Figure 6.7: Sigmoid - plot, equation & range.



Equation	Range
$f(x) = \dfrac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$	$(-1, 1)$

Figure 6.8: Tanh - plot, equation & range.



Equation	Range
$f(x) = x$	$(-\infty, \infty)$

Figure 6.9: Linear - plot, equation & range.

6.8 Neural network modeling

The desired outputs were predicted individually, i.e the output layer contains single neuron (either K_s or MTT), with 3 neurons (representing rake angle, uncut chip thickness and cutting speed) in the input layer. Figure 6.3 shows SNNs built in this work. In both the networks (Figure 6.3a & Figure 6.3b) the number of neurons in the hidden layer were varied from 5 to 25 for all the five activation functions.

Figure 6.4 shows the deep neural networks (DNN) that were built. In both the cases (Figure 6.4a & Figure 6.4b) neurons in the first 2 hidden layers were varied from 5 to 15, whereas the neurons in the third hidden layer were varied from 0 to 3. All the five activation functions were implemented.

One important point to be noted is that, linear activation function was used as default between the output layer and the hidden layer preceding it for all the networks (both SNNs and DNNs).

Later, training process was initiated for all the networks shown in Figures 6.3 and 6.4. A total of 105 neural network architectures were built for each of the SNNs. On the other hand, 800 neural network architectures were built for each of the DNNs. Once the training process was completed the test data sets were fed to determine the network architecture that exhibits the least error in prediction. For this purpose, statistical evaluations have been performed on all the models using mean squared error (MSE), shown in Equation 6.3, and coefficient of determination (R^2), shown in Equation 6.4. In the Equations, 6.3 and 6.4, y_i represents actual output, y_i^p represents ANN predicted output and $\bar{y}$ represents mean of actual outputs. The network architecture with highest R^2 and least MSE on the test data set is concluded to be the suitable network [35].

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^p)^2 \tag{6.3}$$

$$\text{R}^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - y_i^p)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{6.4}$$

CHAPTER 7: ANN RESULTS AND DISCUSSIONS

The ANN modeling involved building both shallow and deep neural networks for predicting maximum tool temperatures and specific cutting forces by utilizing the data generated from finite element simulations.

The objective of the ANN modeling was to determine the suitable neural network architecture and activation function that produced the least error during prediction. This chapter presents the results obtained from the neural networks.

7.1 Prediction of maximum tool temperature

Among the 905 ANN models (SNN + DNN) that were built, the model with the network architecture 3-15-14-3-1 (see Figure 7.1) with the activation function 'ReLU' has the highest R^2 (0.9605) and least MSE (0.00227) on the test data. Figure 7.2 shows the predictions made by this network, it is observed that, the predictions are in close agreement with actual outputs. After examining the plot (Figure 7.2), it can be stated that the network architecture 3-15-14-3-1 is the suitable network for predicting the maximum temperature on the cutting tool.

Table 7.1 presents the performance of the top 50 neural network architectures arranged in increasing order of MSE (or decreasing order of R^2) with respect to the test data set. It is interesting to note that neural networks with ReLU as the activation function have performed well compared to other activation functions.

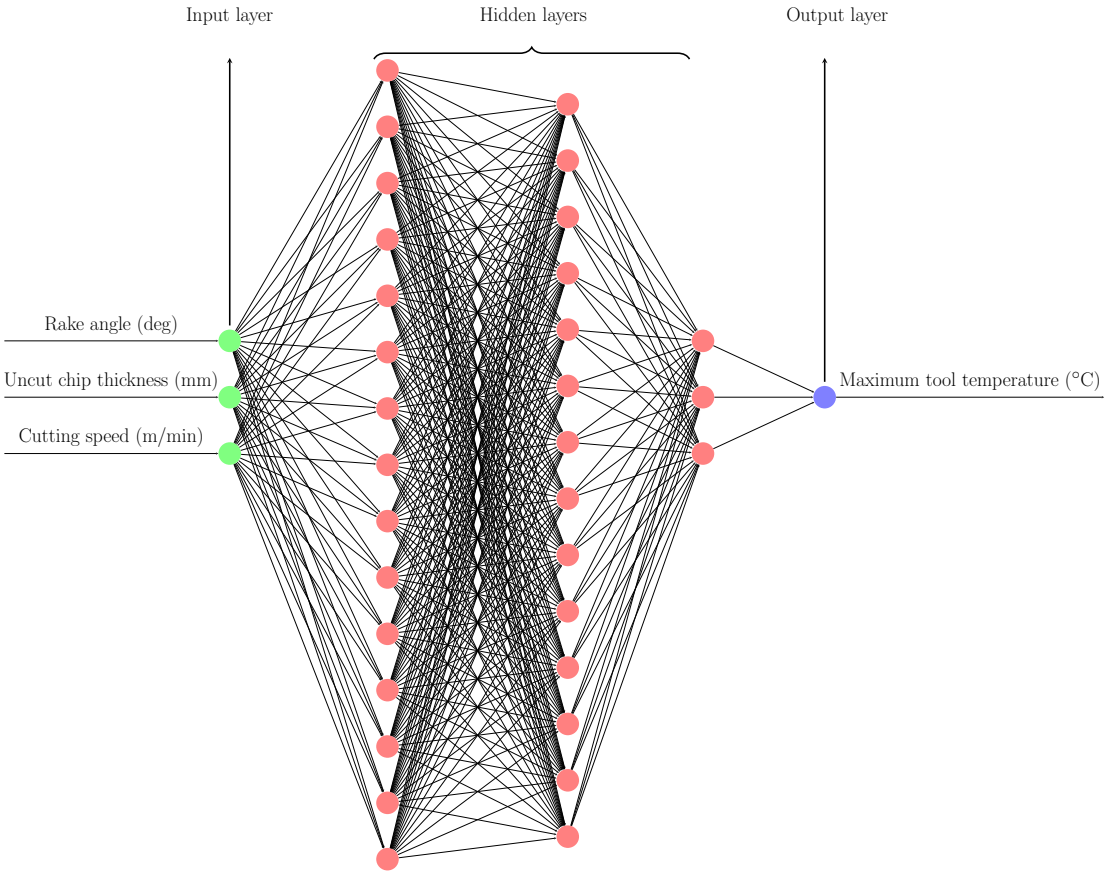


Figure 7.1: Suitable deep neural network with architecture 3-15-14-3-1 for predicting maximum tool temperatures.

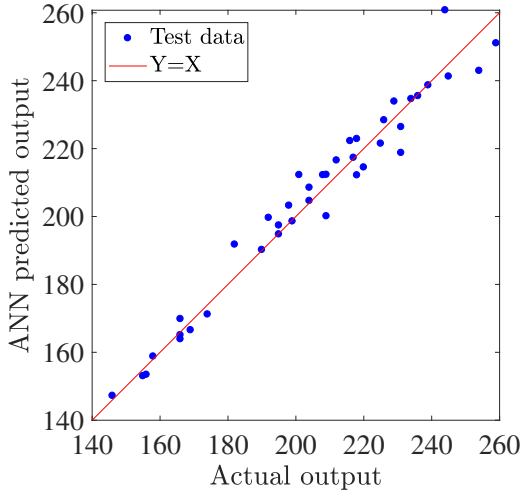


Figure 7.2: Relation between actual values and ANN predicted values for maximum tool temperature.

Table 7.1: Top 50 ANN models for predicting maximum tool temperature.

					Training		Testing	
S No	Activation function	Network type	Network architecture	Hidden layers	MSE	R^2	MSE	R^2
1	ReLU	DNN	3-15-14-3-1	3	0.00113	0.9747	0.00227	0.9605
2	ReLU	DNN	3-14-13-3-1	3	0.00113	0.9748	0.00228	0.9603
3	ReLU	DNN	3-12-13-3-1	3	0.00124	0.9721	0.00235	0.9593
4	ReLU	DNN	3-11-12-3-1	3	0.00144	0.9677	0.00237	0.9588
5	ReLU	DNN	3-11-10-3-1	3	0.00151	0.9662	0.00238	0.9587
6	ReLU	DNN	3-13-13-3-1	3	0.00147	0.9671	0.00248	0.9570
7	ReLU	DNN	3-13-12-0-1	2	0.00140	0.9687	0.00249	0.9568
8	ReLU	DNN	3-15-15-0-1	2	0.00177	0.9603	0.00250	0.9566
9	ReLU	DNN	3-14-14-3-1	3	0.00146	0.9674	0.00256	0.9555
10	ReLU	DNN	3-11-12-0-1	2	0.00166	0.9628	0.00265	0.9540
11	ReLU	DNN	3-14-15-3-1	3	0.00129	0.9712	0.00273	0.9526
12	Tanh	DNN	3-8-9-3-1	3	0.00202	0.9547	0.00282	0.9510
13	Tanh	DNN	3-13-12-3-1	3	0.00218	0.9512	0.00289	0.9499
14	ReLU	DNN	3-7-6-3-1	3	0.00167	0.9627	0.00291	0.9494
15	ReLU	DNN	3-6-6-3-1	3	0.00189	0.9577	0.00294	0.9490
16	ReLU	DNN	3-9-10-3-1	3	0.00168	0.9624	0.00296	0.9486
17	ReLU	DNN	3-11-11-0-1	2	0.00206	0.9539	0.00296	0.9485
18	ReLU	DNN	3-15-14-0-1	2	0.00152	0.9659	0.00297	0.9484
19	Tanh	DNN	3-12-12-3-1	3	0.00221	0.9505	0.00298	0.9483
20	ReLU	SNN	3-23-0-0-1	1	0.00187	0.9582	0.00301	0.9477
21	Tanh	DNN	3-12-13-3-1	3	0.00237	0.9468	0.00305	0.9471
22	Tanh	DNN	3-9-10-2-1	3	0.00223	0.9500	0.00307	0.9467
23	Tanh	DNN	3-8-7-3-1	3	0.00200	0.9553	0.00310	0.9461
24	ReLU	DNN	3-11-10-0-1	2	0.00192	0.9570	0.00314	0.9455
25	ReLU	DNN	3-12-12-3-1	3	0.00183	0.9590	0.00317	0.9449
26	ReLU	DNN	3-9-8-0-1	2	0.00207	0.9536	0.00319	0.9447
27	ELU	DNN	3-15-14-3-1	3	0.00218	0.9512	0.00320	0.9445
28	Tanh	DNN	3-14-13-2-1	3	0.00237	0.9469	0.00322	0.9441
29	Tanh	DNN	3-9-8-3-1	3	0.00241	0.9460	0.00322	0.9441
30	ELU	DNN	3-14-14-3-1	3	0.00233	0.9479	0.00322	0.9440
31	ReLU	SNN	3-24-0-0-1	1	0.00208	0.9534	0.00324	0.9438
32	ReLU	DNN	3-13-12-3-1	3	0.00232	0.9481	0.00325	0.9436
33	ReLU	DNN	3-8-7-3-1	3	0.00210	0.9531	0.00327	0.9432
34	Tanh	DNN	3-15-14-3-1	3	0.00220	0.9509	0.00328	0.9431
35	Tanh	DNN	3-7-8-3-1	3	0.00228	0.9490	0.00329	0.9428
36	ELU	DNN	3-12-12-3-1	3	0.00238	0.9466	0.00330	0.9428
37	ELU	DNN	3-11-10-3-1	3	0.00223	0.9501	0.00330	0.9427
38	ELU	DNN	3-13-12-3-1	3	0.00247	0.9447	0.00331	0.9426
39	Tanh	DNN	3-13-13-3-1	3	0.00256	0.9427	0.00331	0.9425
40	Tanh	DNN	3-11-12-2-1	3	0.00257	0.9426	0.00333	0.9422
41	ReLU	DNN	3-5-5-0-1	2	0.00203	0.9546	0.00333	0.9422
42	Tanh	DNN	3-15-15-2-1	3	0.00251	0.9438	0.00333	0.9421
43	ELU	DNN	3-12-11-3-1	3	0.00233	0.9479	0.00333	0.9421
44	ReLU	DNN	3-14-14-0-1	2	0.00189	0.9576	0.00334	0.9419
45	ELU	DNN	3-15-15-1-1	3	0.00242	0.9458	0.00336	0.9417
46	ELU	DNN	3-11-10-2-1	3	0.00230	0.9486	0.00337	0.9415
47	ReLU	DNN	3-9-8-3-1	3	0.00166	0.9628	0.00337	0.9415
48	ReLU	DNN	3-6-5-0-1	2	0.00173	0.9613	0.00338	0.9414
49	Tanh	DNN	3-11-10-2-1	3	0.00238	0.9467	0.00340	0.9410
50	Tanh	DNN	3-5-5-3-1	3	0.00279	0.9377	0.00340	0.9410

7.2 Prediction of specific cutting force

Among the 905 neural network models, the neural network model with the architecture 3-9-10-0-1 (see Figure 7.3) , 0 indicates that there are no neurons in the third

hidden layer, with 'ReLU' as the activation function has the highest R^2 (0.9419) and least MSE (0.0022) with respect to the test data. After examining the plot (Figure 7.4) which shows the predictions made by the neural network, it can be stated that the network architecture 3-9-10-0-1 is suitable for predicting specific cutting forces.

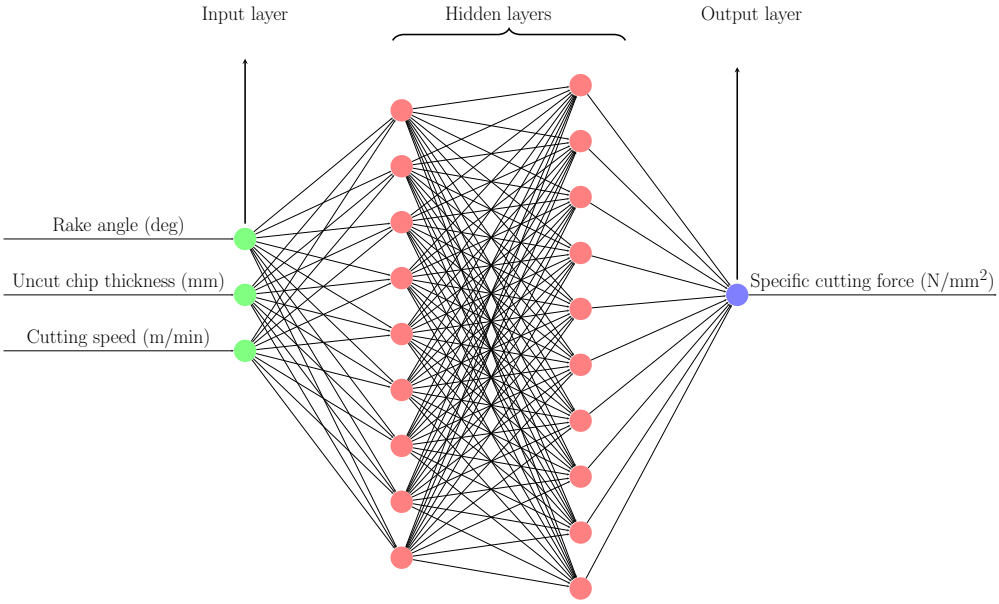


Figure 7.3: Suitable deep neural network with architecture 3-9-10-0-1 for predicting specific cutting forces.

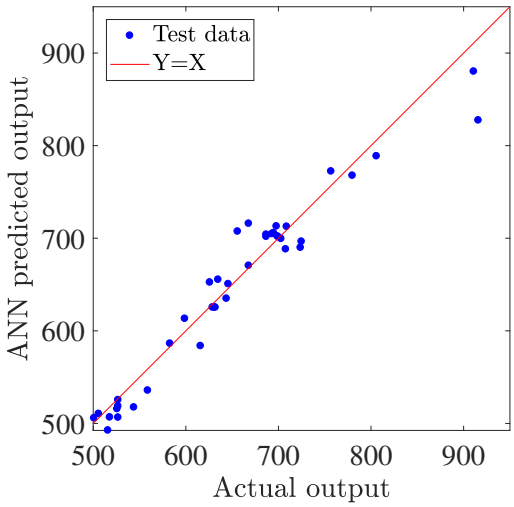


Figure 7.4: Relation between actual values and ANN predicted values for specific cutting force.

Table 7.2: Top 50 ANN models for predicting specific cutting force.

					Training		Testing	
S. No	Activation function	Network type	Network architecture	Hidden layers	MSE	R^2	MSE	R^2
1	ReLU	DNN	3-9-10-0-1	2	0.00294	0.9394	0.00220	0.9419
2	ReLU	DNN	3-8-9-0-1	2	0.00261	0.9463	0.00230	0.9395
3	ELU	DNN	3-13-13-3-1	3	0.00290	0.9403	0.00243	0.9359
4	ReLU	DNN	3-14-15-3-1	3	0.00247	0.9492	0.00246	0.9352
5	ReLU	DNN	3-15-15-0-1	2	0.00242	0.9502	0.00248	0.9348
6	ReLU	DNN	3-13-13-0-1	2	0.00282	0.9420	0.00248	0.9347
7	Tanh	DNN	3-15-14-3-1	3	0.00310	0.9361	0.00253	0.9335
8	Tanh	DNN	3-13-12-3-1	3	0.00304	0.9375	0.00255	0.9329
9	ReLU	DNN	3-5-5-0-1	2	0.00322	0.9338	0.00258	0.9322
10	ELU	DNN	3-6-5-0-1	2	0.00310	0.9362	0.00259	0.9320
11	Tanh	DNN	3-8-9-1-1	3	0.00302	0.9379	0.00259	0.9319
12	ReLU	DNN	3-11-12-3-1	3	0.00264	0.9457	0.00259	0.9319
13	ReLU	DNN	3-13-14-0-1	2	0.00239	0.9507	0.00260	0.9317
14	Tanh	DNN	3-9-8-0-1	2	0.00312	0.9357	0.00262	0.9312
15	Tanh	DNN	3-9-9-3-1	3	0.00313	0.9357	0.00262	0.9310
16	Tanh	DNN	3-9-10-2-1	3	0.00315	0.9353	0.00265	0.9304
17	ELU	DNN	3-9-10-3-1	3	0.00311	0.9359	0.00267	0.9296
18	ELU	DNN	3-13-12-3-1	3	0.00296	0.9392	0.00269	0.9293
19	ReLU	DNN	3-12-13-3-1	3	0.00265	0.9456	0.00269	0.9292
20	Tanh	DNN	3-9-10-3-1	3	0.00322	0.9338	0.00269	0.9292
21	Tanh	DNN	3-13-13-3-1	3	0.00312	0.9357	0.00269	0.9292
22	ReLU	DNN	3-12-12-3-1	3	0.00269	0.9447	0.00271	0.9286
23	ReLU	DNN	3-9-10-3-1	3	0.00258	0.9469	0.00272	0.9284
24	Tanh	DNN	3-11-10-3-1	3	0.00308	0.9367	0.00275	0.9278
25	ReLU	DNN	3-10-11-0-1	2	0.00295	0.9392	0.00276	0.9275
26	Tanh	DNN	3-9-8-3-1	3	0.00376	0.9227	0.00276	0.9274
27	ELU	DNN	3-15-14-3-1	3	0.00304	0.9374	0.00277	0.9270
28	ReLU	DNN	3-14-13-0-1	2	0.00267	0.9450	0.00279	0.9265
29	ReLU	DNN	3-15-14-3-1	3	0.00240	0.9506	0.00280	0.9265
30	Tanh	DNN	3-9-9-2-1	3	0.00332	0.9318	0.00284	0.9252
31	Tanh	DNN	3-14-13-3-1	3	0.00300	0.9382	0.00285	0.9251
32	ReLU	DNN	3-11-11-0-1	2	0.00254	0.9476	0.00285	0.9251
33	Tanh	DNN	3-9-9-1-1	3	0.00343	0.9294	0.00286	0.9247
34	ReLU	DNN	3-6-7-3-1	3	0.00312	0.9359	0.00287	0.9245
35	ELU	DNN	3-11-11-0-1	2	0.00292	0.9400	0.00288	0.9241
36	ReLU	SNN	3-18-0-0-1	1	0.00337	0.9305	0.00291	0.9235
37	ReLU	DNN	3-13-13-3-1	3	0.00276	0.9432	0.00291	0.9234
38	Tanh	DNN	3-8-9-3-1	3	0.00314	0.9354	0.00292	0.9232
39	ReLU	DNN	3-15-14-0-1	2	0.00250	0.9485	0.00293	0.9230
40	Tanh	DNN	3-13-12-0-1	2	0.00330	0.9321	0.00295	0.9224
41	ReLU	DNN	3-8-8-0-1	2	0.00256	0.9472	0.00295	0.9223
42	Tanh	DNN	3-9-9-0-1	2	0.00343	0.9294	0.00295	0.9223
43	ReLU	DNN	3-6-5-0-1	2	0.00312	0.9358	0.00296	0.9221
44	ELU	DNN	3-7-7-0-1	2	0.00303	0.9376	0.00296	0.9220
45	ELU	DNN	3-10-9-3-1	3	0.00309	0.9365	0.00299	0.9213
46	Tanh	DNN	3-8-7-3-1	3	0.00332	0.9316	0.00300	0.9212
47	Tanh	DNN	3-10-9-3-1	3	0.00364	0.9250	0.00300	0.9212
48	Tanh	DNN	3-15-14-2-1	3	0.00312	0.9357	0.00301	0.9209
49	ReLU	DNN	3-14-14-0-1	2	0.00225	0.9537	0.00301	0.9209
50	ReLU	DNN	3-14-14-3-1	3	0.00233	0.9521	0.00302	0.9206

Table 7.2 presents the performance of the top 50 neural network architectures arranged in increasing order of MSE (or decreasing order of R^2) with respect to

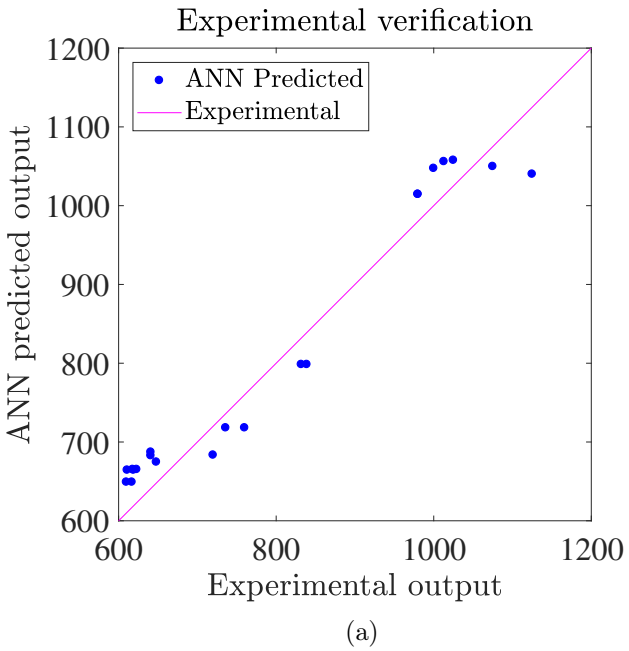
the test data set. It is interesting to note that neural networks with ReLU as the activation function have performed well compared to other activation functions.

7.2.1 Experimental verification

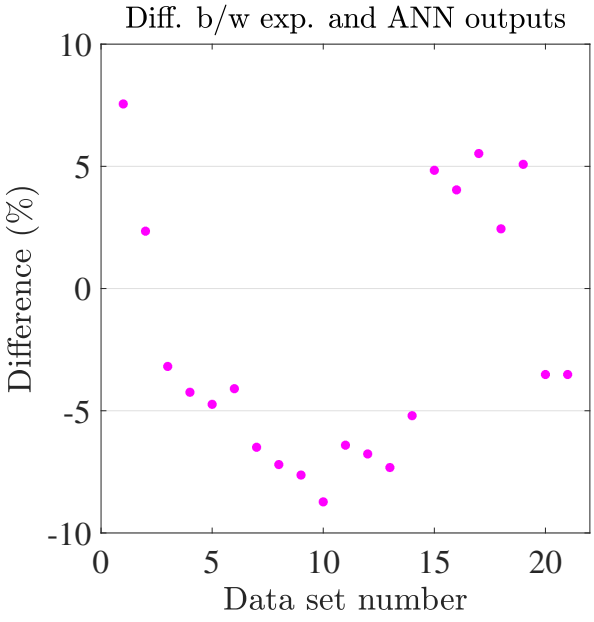
The network architecture 3-9-10-0-1 (Figure 7.3), which was selected for specific cutting force prediction, was further evaluated with the available experimental data. That is, the experimental data sets available in the literature [60],[50],[51],[61] were given as the inputs to this neural network and the corresponding outputs (K_s) were predicted and the difference was calculated.

Figure 7.5a shows the actual experimental outputs and the outputs predicted by this neural network. It can be stated that the selected neural network has made a close prediction of the experimental data set. The corresponding difference in prediction is shown in the Figure 7.5b. The negative difference for certain data sets indicate that the neural network has over-predicted the experimental output.

Specific cutting forces obtained from FEA results, when compared with the physical experimental results for the same cutting parameters (see Figure 4.2), presented in Chapter 4, showed a deviation between 15% to 20%, whereas for the same data sets the ANN predicted outputs (see Figure 7.5b) showed a deviation between -6.5% to -9%. The possible reason for this kind of behavior can be attributed to the accuracy of the neural network selected. The selected neural network's coefficient of determination (R^2) was found to be 0.9419, which is not equal to 1. This could be one of the reasons for the observed differences in deviation. One possible step that can be taken to improve the, R^2, of the neural network is to include more data during the training and testing phases.



(a)



(b)

Figure 7.5: Experimental verification for the network architecture 3-9-10-0-1: Figure (7.5a) shows the ANN predicted outputs for the experimental outputs and Figure (7.5b) shows the corresponding difference in its prediction.

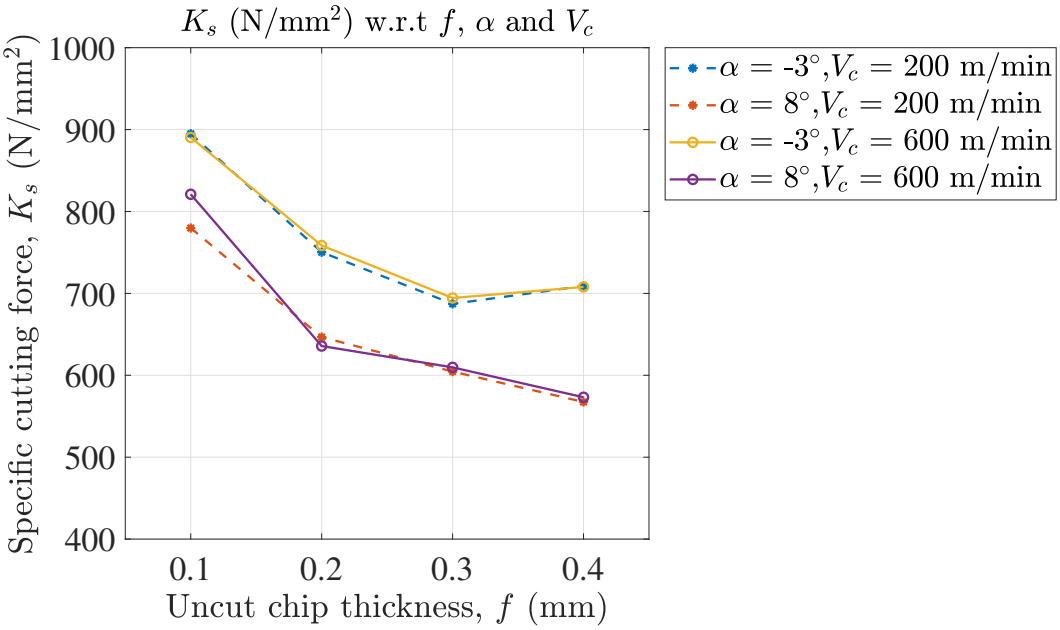
7.3 Sensitivity analysis

Sensitivity studies are extremely important for network designers to predict the effect of input perturbations on the network's output [62]. Sensitivity analysis results tell how likely the outputs based upon the selected model will change on giving new information. The sensitivity of each input is represented by a numerical value, called the sensitivity index. Sensitivity indices are available in several forms. In this work, we are going to focus only on the first-order index. The first-order index measures the contribution to the output variance by a single input alone [63].

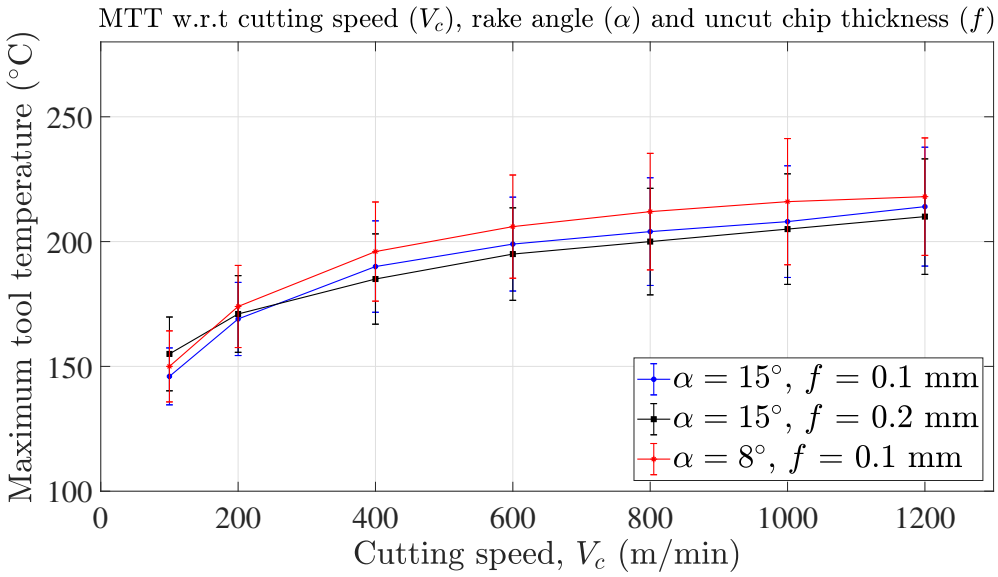
Sensitivity analysis is carried out by using the open source library SALib [63] available for Python programming language. This library is capable of generating the model inputs and computing the sensitivity indices from the model outputs. We got three model inputs(rake angle (α), uncut chip thickness (f) and cutting speed (V_c)). The results obtained after performing sensitivity analysis on the selected neural network architecture, for specific cutting forces (K_s) (Figure 7.3) and maximum tool temperatures (MTT) (Figure 7.1), are shown in the table 7.3. For specific cutting forces both rake angle and uncut chip thickness have impact on the output compared to cutting speed, whereas for maximum tool temperatures cutting speed seems to have more effect on the output compared to both rake angle and uncut chip thickness.

Table 7.3: Sensitivity indices for SCF and MTT.

	First-order indices	
Parameter	K_s (3-9-10-0-1)	MTT (3-15-14-3-1)
Rake angle, (α)	0.5319	0.2206
Uncut chip thickness, (f)	0.4452	0.1316
Cutting speed, (V_c)	0.0009	0.6160



(a)



(b)

Figure 7.6: Specific cutting forces (7.6a) and maximum tool temperature (7.6b) obtained from FEA simulations.

The sensitivity analysis resutls were verified with the results obtained from finite element simulations. Figure 7.6a shows variation in specific cutting forces for rake

angles 8° and -3° during the cutting speeds 200 m/min and 600 m/min for various uncut chip thickness values. It is inferred that specific cutting forces are changing rapidly with the change in uncut chip thickness and rake angle, but they remain alomost same for different cutting speeds.

On the other hand, Figure 7.6b shows the variation in maximum tool temperatures; it can be seen that the temperature is changing rapidly with the increase in cutting speeds but does not show much variation with rake angles (α) and uncut chip thickness (f). Hence, we can conclude that the results obtained from sensitivity analysis are in good agreement with the finite element simulations.

CHAPTER 8: CONCLUSIONS & FUTURE WORK

The work presented a comprehensive analysis of the application of FEM and ANN to predict specific cutting forces and maximum tool temperatures, including a detailed description of modeling orthogonal machining. A total of 196 simulations were performed for various rake angles, uncut chip thickness, and cutting speeds for generating data. In this study, 905 neural network models were built for each specific cutting force and maximum tool temperature prediction. The suitable neural network architecture for predicting specific cutting forces is found to be 3-9-10-0-1, with 'ReLU' as the activation function, whereas for predicting maximum tool temperatures the neural network architecture 3-15-14-3-1, with 'ReLU' as the activation function, was found to be suitable. It was observed that, for both the predictions, 'ReLU' was found to be the suitable activation function. Sensitivity analysis was performed to check the sensitivity of the output with input perturbations, and results revealed that, specific cutting forces are sensitive to both rake angle and uncut chip thickness. On the other hand, maximum tool temperatures were found sensitive to cutting speeds.

The work reveals that the hybrid approach of combining FEM and machine learning to predict specific cutting forces and maximum tool temperatures is effective. The accuracy of the predictions can further be improved by adding more number of data sets during the ANN modeling process.

The proposed approach can be extended for other work materials and manufacturing applications. Additional parameters like stresses, strains, and tool-tip temperatures can be predicted. More advanced training algorithms, such as Nadam and Adamax, can be used along with the application of other activation functions, including, leaky ReLU, PReLU, and Thresholded ReLU.

REFERENCES

[1] T. Childs, K. Maekawa, T. Obikawa, and Y. Yamane, *Metal machining: theory and applications*. Butterworth-Heinemann, 2000.

[2] C. Kılıçaslan, "Modelling and simulation of metal cutting by finite element method," Master's thesis, izmir institute of technology, 2009.

[3] "Abaqus 2017 documentation." http://130.149.89.49:2080/v2016/index.html.

[4] J. P. Patel, *Finite Element Studies of Orthogonal Machining of Aluminum Alloy A2024-T351*. PhD thesis, The University of North Carolina at Charlotte, 2018.

[5] S. Kalpakjian, *Manufacturing engineering and technology*. Pearson Education India, 2001.

[6] S. S. Haykin, S. S. Haykin, S. S. Haykin, K. Elektroingenieur, and S. S. Haykin, *Neural networks and learning machines*, vol. 3. Pearson education Upper Saddle River, 2009.

[7] D. W. Coit, B. T. Jackson, and A. E. Smith, "Static neural network process models: considerations and case studies," *International Journal of Production Research*, vol. 36, no. 11, pp. 2953–2967, 1998.

[8] H. El-Mounayri, H. Kishawy, and J. Briceno, "Optimization of cnc ball end milling: a neural network-based model," *Journal of Materials Processing Technology*, vol. 166, no. 1, pp. 50–62, 2005.

[9] A. P. Markopoulos, D. E. Manolakos, and N. M. Vaxevanidis, "Artificial neural network models for the prediction of surface roughness in electrical discharge machining," *Journal of Intelligent Manufacturing*, vol. 19, no. 3, pp. 283–292, 2008.

[10] E. Usui and T. Shirakashi, "Mechanics of machining-from âdescriptiveâ to âpredictiveâ theory, asme publ," 1982.

[11] K. Iwata, K. Osakada, and Y. Terasaka, "Process modeling of orthogonal cutting by the rigid-plastic finite element method," *Journal of Engineering Materials and Technology*, vol. 106, no. 2, pp. 132–138, 1984.

[12] E. Ceretti, P. Fallböhmer, W. Wu, and T. Altan, "Application of 2d fem to chip formation in orthogonal cutting," *Journal of materials processing technology*, vol. 59, no. 1-2, pp. 169–180, 1996.

[13] H. Bil, S. E. Kılıç, and A. E. Tekkaya, "A comparison of orthogonal cutting data from experiments with three different finite element models," *International Journal of Machine Tools and Manufacture*, vol. 44, no. 9, pp. 933–944, 2004.

[14] J. Stremkowski and J. Carroll, "An orthogonal metal cutting model based on an eulerian finite element method, manufacturing processes," in *Machines and Systems Conference on Production Research and Technology*, vol. 13, p. 261, 1986.

[15] T. Childs and K. Maekawa, "Computer-aided simulation and experimental studies of chip flow and tool wear in the turning of low alloy steels by cemented carbide tools," *Wear*, vol. 139, no. 2, pp. 235–250, 1990.

[16] T. Belytschko, W. K. Liu, B. Moran, and K. Elkhodary, *Nonlinear finite elements for continua and structures*. John wiley & sons, 2013.

[17] A. J. Shih and H. T. Yang, "Experimental and finite element predictions of residual stresses due to orthogonal metal cutting," *International Journal for Numerical Methods in Engineering*, vol. 36, no. 9, pp. 1487–1507, 1993.

[18] H. Sasahara, T. Obikawa, and T. Shirakashi, "The prediction of effects of cutting condition on mechanical characteristics in machined layer," in *Advancement of Intelligent Production*, pp. 473–478, Elsevier, 1994.

[19] T. Marusich and M. Ortiz, "Modelling and simulation of high-speed machining," *International Journal for Numerical Methods in Engineering*, vol. 38, no. 21, pp. 3675–3694, 1995.

[20] L. Olovsson, L. Nilsson, and K. Simonsson, "An ale formulation for the solution of two-dimensional metal cutting problems," *Computers & structures*, vol. 72, no. 4-5, pp. 497–507, 1999.

[21] O. Pantalé, J.-L. Bacaria, O. Dalverny, R. Rakotomalala, and S. Caperaa, "2d and 3d numerical models of metal cutting with damage effects," *Computer methods in applied mechanics and engineering*, vol. 193, no. 39-41, pp. 4383–4399, 2004.

[22] B. McClain, S. A. Batzer, and G. I. Maldonado, "A numeric investigation of the rake face stress distribution in orthogonal machining," *Journal of Materials Processing Technology*, vol. 123, no. 1, pp. 114–119, 2002.

[23] N. Tounsi, J. Vincenti, A. Otho, and M. Elbestawi, "From the basic mechanics of orthogonal metal cutting toward the identification of the constitutive equation," *International Journal of Machine Tools and Manufacture*, vol. 42, no. 12, pp. 1373–1383, 2002.

[24] L. Zhang, "On the separation criteria in the simulation of orthogonal metal cutting using the finite element method," *Journal of Materials Processing Technology*, vol. 89, pp. 273–278, 1999.

[25] İ. Ovalı and A. Mavi, "A study on cutting forces of austempered gray iron using artificial neural networks," *Engineering Science & Technology, an International Journal*, vol. 16, no. 1, 2013.

[26] F. Kara, K. Aslantas, and A. Çiçek, "Ann and multiple regression method-based modelling of cutting forces in orthogonal machining of aisi 316l stainless steel," *Neural Computing and Applications*, vol. 26, no. 1, pp. 237–250, 2015.

[27] P. Asokan, R. R. Kumar, R. Jeyapaul, and M. Santhi, "Development of multi-objective optimization models for electrochemical machining process," *The International Journal of Advanced Manufacturing Technology*, vol. 39, no. 1-2, pp. 55–63, 2008.

[28] A. Al-Ahmari, "Predictive machinability models for a selected hard material in turning operations," *Journal of Materials Processing Technology*, vol. 190, no. 1-3, pp. 305–311, 2007.

[29] S. Kumar and B. Singh, "Ascertaining of chatter stability using wavelet denoising and artificial neural network," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 233, no. 1, pp. 39–62, 2019.

[30] R. Thirumalainambi and J. Bardina, "Training data requirement for a neural network to predict aerodynamic coefficients," in *Independent Component Analyses, Wavelets, and Neural Networks*, vol. 5102, pp. 92–104, International Society for Optics and Photonics, 2003.

[31] S. Dahbi, L. Ezzine, and H. EL Moussami, "Modeling of cutting performances in turning process using artificial neural networks," *International Journal of Engineering Business Management*, vol. 9, p. 1847979017718988, 2017.

[32] H. Cherukuri, E. Perez-Bernabeu, M. A. Selles, and T. Schmitz, "Machining chatter prediction using a data learning model," *Journal of Manufacturing and Materials Processing*, vol. 3, no. 2, 2019.

[33] I. Korkut, A. Acır, and M. Boy, "Application of regression and artificial neural network analysis in modelling of tool–chip interface temperature in machining," *Expert Systems with Applications*, vol. 38, no. 9, pp. 11651–11656, 2011.

[34] A. E. Tumer and S. Edebali, "An artificial neural network model for wastewater treatment plant of konya," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 3, no. 4, pp. 131–135, 2015.

[35] S. Tasdemir, "Artificial neural network model for prediction of tool tip temperature and analysis," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 6, no. 1, pp. 92–96, 2018.

[36] S. Tasdemir, "Artificial neural network based on predictive model and analysis for main cutting force in turning," *Energy Education Science and Technology Part A-Energy Science and Research*, vol. 29, no. 2, pp. 1471–1480, 2012.

[37] F. J. Pontes, J. R. Ferreira, M. B. Silva, A. P. Paiva, and P. P. Balestrassi, "Artificial neural networks for machining processes surface roughness modeling," *The International Journal of Advanced Manufacturing Technology*, vol. 49, no. 9-12, pp. 879–902, 2010.

[38] M. Correa, C. Bielza, and J. Pamies-Teixeira, "Comparison of bayesian networks and artificial neural networks for quality detection in a machining process," *Expert systems with applications*, vol. 36, no. 3, pp. 7270–7279, 2009.

[39] A. P. Markopoulos, *Finite element method in machining processes*. Springer Science & Business Media, 2012.

[40] M. H. Ali, M. Ansari, B. A. Khidhir, B. Mohamed, and A. Oshkour, "Simulation machining of titanium alloy (ti-6al-4v) based on the finite element modeling," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 36, no. 2, pp. 315–324, 2014.

[41] F. Harewood and P. McHugh, "Comparison of the implicit and explicit finite element methods using crystal plasticity," *Computational Materials Science*, vol. 39, no. 2, pp. 481–494, 2007.

[42] T. Mabrouki, F. Girardin, M. Asad, and J.-F. Rigal, "Numerical and experimental study of dry cutting for an aeronautic aluminium alloy (a2024-t351)," *International Journal of Machine Tools and Manufacture*, vol. 48, no. 11, pp. 1187–1197, 2008.

[43] X. Teng and T. Wierzbicki, "Evaluation of six fracture models in high velocity perforation," *Engineering Fracture Mechanics*, vol. 73, no. 12, pp. 1653–1678, 2006.

[44] N. Zorev, "Inter-relationship between shear processes occurring along tool face and shear plane in metal cutting," *International research in production engineering*, vol. 49, pp. 143–152, 1963.

[45] P. Wallace and G. Boothroyd, "Tool forces and tool-chip friction in orthogonal machining," *Journal of Mechanical Engineering Science*, vol. 6, no. 1, pp. 74–87, 1964.

[46] X. Yang and C. R. Liu, "A new stress-based model of friction behavior in machining and its significant impact on residual stresses computed by finite element method," *International Journal of Mechanical Sciences*, vol. 44, no. 4, pp. 703–723, 2002.

[47] T. Hsu, "A study of the normal and shear stresses on a cutting tool," *Journal of engineering for industry*, vol. 88, no. 1, pp. 51–61, 1966.

[48] G. R. Johnson and W. H. Cook, "Fracture characteristics of three metals subjected to various strains, strain rates, temperatures and pressures," *Engineering fracture mechanics*, vol. 21, no. 1, pp. 31–48, 1985.

[49] J. Patel and H. P. Cherukuri, "Chip morphology studies using separate fracture toughness values for chip separation and serration in orthogonal machining simulations," in *ASME 2018 13th International Manufacturing Science and Engineering Conference*, pp. V002T04A031–V002T04A031, American Society of Mechanical Engineers, 2018.

[50] M. Asad, F. Girardin, T. Mabrouki, and J.-F. Rigal, "Dry cutting study of an aluminium alloy (a2024-t351): a numerical and experimental approach," *International Journal of Material Forming*, vol. 1, no. 1, pp. 499–502, 2008.

[51] M. Madaj and M. Píška, "On the sph orthogonal cutting simulation of a2024-t351 alloy," *Procedia CIRP*, vol. 8, pp. 152–157, 2013.

[52] S. Kouadri, K. Necib, S. Atlati, B. Haddag, and M. Nouari, "Quantification of the chip segmentation in metal machining: Application to machining the aeronautical aluminium alloy aa2024-t351 with cemented carbide tools wc-co," *International Journal of Machine Tools and Manufacture*, vol. 64, pp. 102–113, 2013.

[53] V. P. Astakhov and S. Shvets, "The assessment of plastic deformation in metal cutting," *Journal of Materials Processing Technology*, vol. 146, no. 2, pp. 193–202, 2004.

[54] T. D. Marusich, "Effects of friction and cutting speed on cutting force," in *Proceedings of ASME Congress*, pp. 11–16, 2001.

[55] D. Parle, R. K. Singh, and S. S. Joshi, "Modeling of specific cutting energy in micro-cutting using sph simulation," in *IWMF2014, 9th international workshop on microfactories*, pp. 121–126, 2014.

[56] D. A. Lucca, Y. Seo, and R. Komanduri, "Effect of tool edge geometry on energy dissipation in ultraprecision machining," *CIRP annals*, vol. 42, no. 1, pp. 83–86, 1993.

[57] N. Ketkar, "Introduction to keras," in *Deep Learning with Python*, pp. 97–111, Springer, 2017.

[58] F. Chollet, "Keras documentation," *keras. io*, 2015.

[59] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[60] S. Atlati, B. Haddag, M. Nouari, and M. Zenasni, "Analysis of a new segmentation intensity ratio âsirâ to characterize the chip segmentation process in machining ductile metals," *International Journal of Machine Tools and Manufacture*, vol. 51, no. 9, pp. 687–700, 2011.

[61] S. Kobayashi, R. Herzog, D. Eggleston, and E. Thomsen, "A critical comparison of metal-cutting theories with new experimental data," *Journal of Engineering for Industry*, vol. 82, no. 4, pp. 333–347, 1960.

[62] X. Zeng and D. S. Yeung, "Sensitivity analysis of multilayer perceptron to input and weight perturbations," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1358–1366, 2001.

[63] J. Herman and W. Usher, "Salib: An open-source python library for sensitivity analysis," *The Journal of Open Source Software*, vol. 2, jan 2017.