Deusto
Universidad de Deusto
Deustuko Unibertsitatea
University of Deusto

TESIS DOCTORAL

Programa de Doctorado en
Engineering for the Information Society and Sustainable Development

NOMBRE APELLIDO
Dmitrii Viatkin

Curso 2019 /2022

TÍTULO

Advanced artificial intelligence methods applied to societal challenges in biomedical engineering

This dissertation examines the possibilities of using artificial intelligence methods in biomedical engineering tasks, at the edge of medicine and engineering. The possibilities of analysis, development, use and interpretation of artificial intelligence algorithms in applied problems for sustainable development of society, medical and industrial development are considered. The dissertation consists of two case studies conducted in Spain and Russia, each using a different methodology and approach to analysis.

The first case study explores the application of deep learning to the task of measuring the position of patients' fingers in multiple sclerosis. Tracking the limited degree of mobility of the fingers on the hand can be used as a marker to characterize the course of multiple sclerosis and the success of the treatment prescribed. The objectives of this case study were to review and analyze the literature on the various methods available for assessing finger position, to collect and prepare data for a single camera-based computer vision system designed to detect finger position, and to train and test a neural network based on a neural network for assessing finger position.

The second case study explores the potential of deep learning methods for materials analysis and the possibility of applying them for biomedical purposes. This case study explores the potential of neural networks to analyze the properties and structure of materials with different amounts of data and different representations. The neural network prediction of critical superconductivity temperature for materials based on their chemical formula was considered. The prediction of the reduced glass transition temperature of metal alloys based on a neural network was considered. The prediction of material composition based on the required physical parameters for cellulosic materials was considered. The use of generative-adversarial networks to predict the properties and composition of metal alloys based on incomplete material information with an acceptable range of predicted parameters was considered. The second case study demonstrates the development of the idea of applying neural networks to materials problems, from predicting a single parameter from a chemical formula, to predicting physical parameters and material composition based on incomplete data.

UNIVERSIDAD DE DEUSTO

ADVANCED ARTIFICIAL INTELLIGENCE METHODS APPLIED TO SOCIETAL CHALLENGES IN BIOMEDICAL ENGINEERING

Doctoral thesis by

Dmitrii Viatkin

Directors

BEGONA GARCIA-ZAPIRAIN SOTO
AMAIA MENDEZ ZORRILLA

Bilbao, July 2022

UNIVERSIDAD DE DEUSTO

ADVANCED ARTIFICIAL INTELLIGENCE METHODS APPLIED TO SOCIETAL CHALLENGES IN BIOMEDICAL ENGINEERING

Doctoral thesis by: DMITRII VIATKIN

Industrial Ph.D. between

University of Deusto, Spain

Northern (Arctic) Federal University named after MV

Lomonosov, Russia

Director:

BEGONA GARCIA-ZAPIRAIN SOTO
AMAIA MENDEZ ZORRILLA

Doctoral student Directors

Bilbao, July 2022

DECLARATION

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and acknowledgements.

Dmitrii Viatkin

June 2022

ABSTRACT

In recent years, artificial intelligence and machine learning algorithms have been increasingly used not only in scientific, but also in applied and societal fields. This is due to the development of computing power, technological development and the development of the algorithms used. Developed algorithms of artificial intelligence begin to be introduced also in medicine, engineering, bioengineering, biomedical and other frontier areas, where different knowledge areas touch each other. Development and training of artificial intelligence algorithms for solving problems, which are at the interaction boundary of different fields of science, can improve the quality of interaction between experts from different fields, expand the frontiers of knowledge and solve applied problems in the areas under study.

This dissertation examines the possibilities of using artificial intelligence methods in biomedical engineering tasks, at the edge of medicine and engineering. The possibilities of analysis, development, use and interpretation of artificial intelligence algorithms in applied problems for sustainable development of society, medical and industrial development are considered. The dissertation consists of two case studies conducted in Spain and Russia, each using a different methodology and approach to analysis.

The first case study explores the application of deep learning to the task of measuring the position of patients' fingers in multiple sclerosis. Tracking the limited degree of mobility of the fingers on the hand can be used as a marker to characterize the course of multiple sclerosis and the success of the treatment prescribed. The objectives of this case study were to review and analyze the literature on the various methods available for assessing finger position, to collect and prepare data for a single camera-based computer vision system designed to detect finger position, and to train and test a neural network based on a neural network for assessing finger position.

The second case study explores the potential of deep learning methods for materials analysis and the possibility of applying them for biomedical purposes. This case study explores the potential of neural networks to analyze the properties and structure of materials with different amounts of data and different representations. The generation of materials based on a number of incomplete parameters with limited data has been studied. Algorithms for processing different types of material data representations and their parameters have been studied. In this case study, the following tasks were accomplished: literature review and analysis on various available material analysis methods, collection and preparation of data for a material analysis system with different structures and parameters, training and testing the neural network on the collected data. The neural network prediction of critical superconductivity temperature for materials based on their chemical formula was considered. The prediction of the reduced glass transition temperature of metal alloys based on a neural network was considered. The prediction of material composition based on the required physical parameters for cellulosic materials was considered. The use of generative-adversarial networks to predict the properties and composition of metal alloys based on incomplete material information with an acceptable range of predicted parameters was considered. The second case study demonstrates the development of the idea of applying neural networks to materials problems, from predicting a single parameter from a chemical formula, to predicting physical parameters and material composition based on incomplete data.

Keywords: neural network; algorithm; multiple sclerosis; material; model training; analysis

ACKNOWLEDGMENT

There are not enough words to express my gratitude to the many people and institutions that supported me during my graduate studies.

First of all, I would like to express my deepest gratitude to my supervisors Professor Amaia Mendez Zorrilla and Begona Garcia-Zapirain for their support, exchange of experiences and fruitful discussions. Without your support, this dissertation work would not have been possible.

Thanks to the team of evida Laboratory, University of Deusto, Spain, for their support and help with the research.

Thanks to Enrique Onieva Caracuel, University of Deusto, Spain, for help with organizational moments.

Thanks to Ilya Mayorov and Maxim Zakharov, Northern Arctic Federal University, Russia for their research assistance.

Thanks to Zarubin Lyubov and Fedorovna Natalia, Northern Arctic Federal University, Russia, for help with organizational issues.

I thank my family for their moral support during the whole work on the dissertation.

Dmitrii Viatkin

June 2022
Bilbao, Spain

This thesis is dedicated to my family,

for all their love and unconditional support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

In modern society, scientific, technical and societal processes are actively progressing. In the process of its progress, these processes create new opportunities and new challenges (Claps et al., 2015; D'Oca et al., 2018). The faster society process, the more opportunities and challenges appear (Cath, 2018). The use of new opportunities to solve challenges is an integral part of human society development.

The active evolution of computing power, artificial intelligence and machine learning in the 21st century has provided many opportunities to address scientific, technical and societal challenges (Claps et al, D'Oca et al, 2018; Gerke et al, 2020; Glikson et al, 2020). But active development has also created new challenges (Price et al, 2019; Dignum, 2018; Mutascu, 2021). The scientific, technical, and societal spheres of human activity are closely linked. Solving one challenge can help solve others in a neighboring field or create new ones.

The development of artificial intelligence can help automate routine operations, simplify the creation of new devices, medicines, materials, and improve the quality of human life (Prentice et al, 2019; Amann et al, 2020). However, any change in human activity generates challenges: moral, legislative, technical and others.

For example, the automation of workplaces may simplify human labor, but it entails a reduction in the number of jobs, which may lead to an increase in unemployment and social tensions (Price et al, 2019; Mutascu, 2021; Prentice et al, 2019).

The use of artificial intelligence for medical tasks could make high-quality medicine widely available (Cath, 2018; D'Oca et al, 2018; Hamet et al, 2017). However, there is an issue of trust in artificial intelligence systems and the question of the distribution of responsibility in case of unforeseen situations (Glikson et al, 2020; Hamet et al, 2017; Keskinbora, 2019).

However, the refusal to introduce new technologies into people's lives does not cancel out the increasing challenges that arise in the social sphere. New challenges are constantly arising. If technology is not implemented, there will be stagnation and an explosion of growing challenges (Thomas et al, 2022; Alexis, 2021).

This dissertation examines the application of artificial intelligence methods to societal challenges in various spheres of human life. The dissertation considers two main cases of study.

Case study 1. Deep Learning Techniques applied to predict and measure finger movement for the patients with multiple sclerosis.

The possibilities, perspectives, and limitations of applying computer vision algorithms based on neural networks in the medical sphere for tracking mobility limitations of patients with multiple sclerosis are considered. Multiple sclerosis is a complex neurodegenerative disease where people progressively get more limited in movement and cannot move properly. Tracking patient motion limitations is now a complex process that requires specialized medical equipment. One of ways of tracking motion limitations based on visual observations. Computer vision techniques can be used to improve existing medical methods of evaluating the patient's movement skills are based on visual observation. Tracking changes in hand mobility in multiple sclerosis is challenging because accurate hand mobility measurement requires at least measurement of at least 12 joint angles for one hand.

Case study 2. A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications

The possibilities, prospects, and limitations of using artificial intelligence algorithms to analyze and design the properties of materials and their composition are discussed in this case study. New materials are used in scientific and societal fields of human activity. However, the design and analysis of new materials is a complex process. The properties of materials are highly dependent on their structure, composition, and chemical formula. Neural networks can help simplify the process of developing new materials. This case study looked at neural network prediction of the critical superconductivity temperature for materials based on their chemical formula. Prediction of material composition based on required physical parameters for cellulosic products was also considered. The use of generative-adversarial networks for predicting the properties and composition of metal alloys based on incomplete information about the material with the allowable range of predicted parameters was also considered.

This dissertation focuses on the Sustainable Development Goals 2030. The Sustainable Development Goals (SDGs) are a collection of 17 global goals designed to be a «blueprint to achieve a better and more sustainable future for all». The SDGs, set in 2015 by the United Nations General Assembly and intended to be achieved by the year 2030, are part of UN Resolution 70/1, the 2030 Agenda. Figure 1.1 shows the Sustainable Development Goals 2030 plan.

Figure 1.1. The Sustainable Development Goals 2030 plan.

Case study 1 “Deep Learning Techniques applied to predict and measure finger movement for the patients with multiple sclerosis” is related to goal 3 of the Sustainable Development Goals 2030. The details of this case study's relevance to the goal 3 of the Sustainable Development Goals 2030 are shown in section 2.2.1. Sustainable Development Goals 2030. Good health and well-being.

Case study 2 “A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications” is related to the goal 9 of the Sustainable Development Goals 2030. The details of this case study's relevance to the goal 9 of the Sustainable Development Goals 2030 are shown in section 2.2.2. Sustainable Development Goals 2030. Industry, innovation and infrastructure.

1.1 Research hypothesis and objectives

The following main hypotheses are highlighted and studied in the process of research:

1. Artificial intelligence methods have a wide range of applications and can solve different societal challenges in biomedical field.
2. Deep learning algorithms can automatically measure with high accuracy the angles value of the hand in one image
3. Neural networks can be used for the material analysis for the prediction of the properties of materials based on their composition and prediction the materials composition based on the material composition

This hypothesis has been implemented in explored cases of study.

The main aim of this dissertation is to research the possibilities of development and implementation of the advanced artificial intelligence methods for societal challenges in biomedical engineering, considering the features of their software and hardware implementation.

Case study 1. Deep Learning Techniques applied to predict and measure finger movement for the patients with multiple sclerosis.

Analysis of the problem presented in the previous section led to the following research questions (RQ).

RQ 1. Deep learning algorithms can automatically measure with high accuracy the angles value of the hand in one image

RQ 2. The use of non-standard ways of data augmentation allows to expand the expand the possibilities of applying the neural network and increase its stability

Based on the hypothesis and the research questions posed, the goal of this dissertation for this case study is to develop a system for predicting and measuring finger movements in patients with multiple sclerosis based on deep learning and computer vision algorithms.

Hypotheses 1 and 2 are considered in this case study. This case study examines the possibilities of developing and implementing advanced artificial intelligence methods for the societal challenges from goal 3 “Good health and well-being” of the “Sustainable Development Goals 2030”.

To reach the main aim of this dissertation and hypothesis testing, the following specific objectives (SO) need to be fulfilled.

SO 1. Determine the possibilities of different computer vision algorithms to solve the problem of measuring the joint angles of fingers.

SO 2. Collect data for training the computer vision system to measure the angles of the joint angles of fingers.

SO 3. Develop and software implementation of computer vision algorithm for measuring the joint angles of fingers. Training neural network model.

SO 4. Create of the test version of the program. Debugging and testing the developed algorithm. To draw conclusions about the system quality.

SO 5. Create of the final real time implementation of the program based on the developed algorithm.

Case study 2. A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications

Analysis of the problem presented in the previous section led to the following research questions (RQ).

RQ 1. Neural networks can predict the properties of materials based on their chemical composition

RQ 2. Neural networks can predict the composition of materials based on their physical and chemistry properties

RQ 3. Neural networks can be used for generation of the material properties and composition description and evaluation of the generated descriptions

Hypotheses 1 and 3 are considered in this case study. This case study examines the possibilities of developing and implementing advanced artificial intelligence methods for the societal and industrial challenges from goal 9 “Industry, innovation and infrastructure” of the “Sustainable

Development Goals 2030".

Based on the hypothesis and the research questions posed, the goal of this dissertation for this case study is to develop a system for predicting and measuring properties and composition of materials based on incomplete information about the material.

To reach the main aim of this dissertation, the following specific objectives (SO) need to be fulfilled.

SO 1. Analyze various approaches to materials analysis

SO 2. Collect and prepare data for analyzing the properties of materials

SO 3. Develop an algorithm and train a neural network to predict target parameters of materials

SO 4. Develop a program to use the developed trained neural network model

SO 5. Test and debug the developed program

This case study was conducted in several steps.

In step 1, neural network prediction of critical superconductivity temperature for materials based on their chemical formula was considered.

Step 2 considered the prediction of the reduced glass transition temperature of metal alloys based on a neural network.

Step 3 considered the prediction of material composition based on the required physical parameters for cellulosic materials.

Step 4 considered the use of generative-adversarial networks to predict the properties and composition of metal alloys based on incomplete material information with an acceptable range of predicted parameters.

1.2 Scientific and social impact and contribution

In addition to the specific goals mentioned above, the following goals were set to contribute to the research community. These two goals were identified based on the nature of the topic of this thesis.

1. To maximize the scholarly contribution of this thesis with the publication of several articles in scholarly, peer-reviewed conferences and journals relevant to the proposed topic.

2. To ensure maximum clarity and reproducibility of the various methodologies and data processing algorithms used. This will allow future developers and researchers to implement, improve and/or replicate all of the various research issues addressed in this document.

These contributions are part of the thesis, and they have been included in various sections of this dissertation.

The research examines the social aspect of the impact of the issues under study and the systems being developed. In the case studies described above, the reasons and possibilities of implementation of artificial intelligence systems to solve practical issues from the social and technical point of view are considered. This dissertation provides a description of artificial intelligence systems under development with a rationale for the chosen solutions and their alternatives.

The research also considers the technological and industrial aspect of the impact of the issues being studied and the systems being developed. The possibility of using the results in the biomedical sphere is considered. A description and justification of the decisions made in the study and their alternatives. A description of the process of development and training of neural networks for reproducibility of the research carried out is given. The prospects for the development and implementation of the proposed ideas are given, considering the available technology.

1.3 Research methodology

There are several important steps in this research. Figure 1.2 shows stages of the described PhD research.

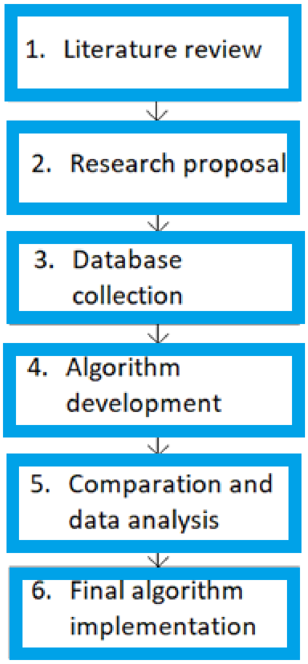


Figure 1.2. Stages of the described PhD research

These stages are similar for each case study but have specific features for each case study. The features of each stage for each case study are listed below.

Case study 1. Deep Learning Techniques applied to predict and measure finger movement for the patients with multiple sclerosis.

1. Literature review.

At this stage the sources related to multiple sclerosis, its symptomatology and methods of treatment were considered. The existing approaches to the implementation of artificial intelligence and computer vision algorithms for medical tasks were also studied.

2. Research proposal

Detecting the symptoms of multiple sclerosis and tracking their development requires expensive and specialized equipment. One important symptom is limitation of limb mobility in patients. Measuring the degree of mobility limitation in multiple sclerosis can allow the progression of the disease and the effectiveness of treatment to be monitored. The effect of mobility limitation can be seen most prominently in the hand because of the many moving joints.

For further research in this case study, the task of measuring the position of the hand was chosen. Measuring hand position in the boundary positions available to the patient may allow measurement of hand mobility limitations in multiple sclerosis and similar diseases.

Measuring hand position requires sophisticated, expensive equipment. In this case study, the possibility of using artificial intelligence to solve the chosen problem was explored. The articles examined approaches based on sensor systems and 3D depth cameras.

Approaches to measure geometric parameters of three-dimensional objects using mathematical models of photogrammetry and 3D cameras have already been developed. However, human perception allows to perceive 3D objects and their geometric parameters with one eye. Based on the reviewed materials, it has been suggested that a single camera without 3D vision is sufficient to measure the hand position.

Mathematical and algorithmic description of this model by classical methods would be extremely difficult. However, human perception is capable of this, so the use of algorithms based on artificial neural networks to analyze the camera image was proposed. Case study "Deep Learning Techniques applied to predict and measure finger movement for the patients with multiple sclerosis" has been selected.

3. Database collection

Open data sources with a description of the hand position are designed to find the hand position in the frame, build a 3D model of the hand based on multiple cameras or gesture recognition. No datasets for hand position recognition with joint position parameters are available in the public domain. The dataset was assembled independently.

The attempt to collect high-precision data using sensors was unsuccessful, because the neural network focused on the position of the sensors on the hand when analyzing the images. Such a neural network did not work correctly on images of the hand without sensors.

A simulation approach was chosen to collect the data. A highly polygonal 3D model of the hand with customizable parameters and surroundings was developed. A camera captured the position of the hand from different angles.

The position of 12 joints of the hand, 3 each for the index, middle, ring, and little finger, was chosen to determine the degree of mobility limitation.

This approach allowed obtaining a variety of high-quality images of the hand in different positions with different parameters. This dataset was used in the following work.

4. Algorithm development

In the process of determining the position of the hand based on a single image, various computer vision methods were considered. Using classical computer vision was unsuccessful; the system under development was unstable and slow. A neural network approach was proposed. The neural network was trained on previously obtained data This approach was successful and made it possible to determine the position of hand joint angles with a mean absolute error of 4.802 degrees for high-quality images and 5.301 for low-quality images.

5. Comparation and data analysis

The developed system based on a single camera and a neural network to analyze the obtained images was compared with other approaches, based on sensors and 3D cameras. The system was tested and options for its possible improvement were suggested. In the process of testing the first versions, it was found that correct results are possible only for high-quality images. Work on images with low quality became possible after adding images with graphic deformations and artifacts to the training sample.

6. Final algorithm implementation

The final version of the neural network was optimized to run in real time and consume minimal computational resources. Adding graphical deformations and artifacts to the images reduced the dependence of the quality of hand position recognition on the camera parameters. The system can be used for medical purposes under almost any conditions and on almost any equipment. A distinctive aspect of the system is its component cost, ease of maintenance.

A neural network model is used for analyzing images coming from one camera. The result of the neural network model analyzing is the measurement of the joint angles of the hand on the image coming from the camera. This case study focuses on the development of a system for measuring finger joint angles based on camera image and is intended for work within the field of medicine to track the movement and limits of hand mobility in multiple sclerosis. Measuring changes in hand mobility allows the progress of the disease and its treatment process to be monitored.

A static RGB camera without depth vision was used in the system developed, with the system receiving only the image from the camera and no other input data. The case study focuses on the analysis of each image in the video stream independently of other images from that stream, and 12 measured hand parameters were chosen as follows: 3 joint angles for the index finger, 3 joint angles for the middle finger, 3 joint angles for the ring finger, and 3 joint angles for the pinky finger.

The final neural network used for image analysis was a modernized neural network based on MobileNetV2, which obtained the best mean absolute error value of 4.757 degrees. Additionally, the mean square error was 67.279 and the root mean square error was 8.202 degrees. This neural network analyzed a single image from the camera without using other sensors.

Case study 2. A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications

1. Literature review

Articles on algorithms and ways of using artificial intelligence were analyzed. It was found that neural networks are able to analyze data based on physical parameters with high quality and effective.

Existing systems for analyzing the properties of materials and the degree of applicability of artificial intelligence algorithms in these systems were analyzed.

Sources describing the practical application of new materials in various scientific, societal, medical, and technological aspects of human life were also considered at this stage.

2. Research proposal

Material development is an important part of the sustainable development of society. The properties of various materials are based on physical laws. The properties of materials are determined by the properties of their components.

It has been suggested that the analysis of material components using artificial intelligence algorithms may allow prediction of their properties.

The use of neural networks in predicting the properties and composition of materials can reduce the cost and complexity of developing new materials, which will improve the scientific, social, medical, and technological aspects of human life.

3. Database collection

This case study had several steps. For each of these steps, data were collected and further processed to achieve the objectives.

4. Algorithm development

This case study had several steps. In step 1, the neural network was trained to predict the superconductivity temperature of materials based on their chemical formula. In step 2, the neural network was trained to predict the reduced glass transition temperature of metal alloys based on a neural network. In step 3, for cellulosic materials, the material composition was predicted based on the required properties. In step 4, a generative-adversarial neural network was trained to generate materials with specified structural and physical parameters in specified ranges based on incomplete data on material composition and properties.

5. Comparation and data analysis

Developed systems are compared with other developed systems. Developed systems are analyzed and decisions are made to improve them.

6. Final algorithm implementation

The neural networks developed, trained and optimized for the task are tested on the available data. The results are recorded and published.

In step 1 neural networks for predicting the critical superconductivity temperature of materials based on their chemical formula has been developed. Recurrent neural networks including long-term short-term memory layers and neural networks based on one-dimensional convolutional layers are used for data analysis. The proposed model is an ensemble of pre-trained neural network architectures for predicting the critical temperature of superconductors based on their chemical formula. The architecture of the seven pre-trained neural networks is based on long-term short-term memory layers and convolution layers. The final ensemble uses six neural networks: one LSTM-based network and four convolutional neural networks, and one ensemble of convolutional neural networks. The LSTM neural network and convolutional neural network were trained over 300 epochs. An ensemble of models was trained in 20 epochs. All neural networks are trained in two phases. In both phases Adam optimizer was used. The first stage was trained using Mean Absolute Error loss function with learning rate of the optimizer equal to 0.001. The second stage was trained with Mean Squared Error loss function with learning rate equal to 0.0001. The final ensemble is trained with a learning rate equal to 0.00001. The final ensemble model has the following accuracy values: Mean Absolute Error is 4.068, Mean Squared Error is 67.272, and Coefficient of Determination is 0.923. The final model can predict the critical temperature for the chemical formula with an accuracy of 4.068 degrees.

In step 2 the neural network system for predicting the glass transition temperature of metal alloys Trg based on recurrent neural network has been developed. The developed system can predict the reduced glass transition temperature Trg of metal alloys with high accuracy based on the analysis of their chemical formula. The accuracy was evaluated by 3 metrics: MSE, RMSE, MAE. The values obtained are as follows: MSE - 0.000678, RMSE - 0.0260, MAE - 0.01835.

In step 3 a neural network to analyze the composition of paper-like materials based on specified chemical and physical parameters has been developed. Five models are trained and predict 5 parameters independently. Each model has 15 input parameters. Each model has its own predicted parameter describing the physical and chemical composition of the material approximating the properties of the given input physical and chemical input parameters. On the available data for calculation of concentrations of different particle sizes: nano solid fibers NSF-0,1; micro solid fibers MSF-0,25; macro solid fibers MSF-0,4; ultra-solid fibers USF-0,6 the values of average absolute error of calculation of concentrations equal to 4.1%; 5.3%; 23.4%; 14% respectively were obtained. For the calculation of Al2(SO4)3 concentrations, the value of the mean absolute error is 1.5%.

In step 4 of a generative-adversarial neural network for generating metal alloy compounds with given parameters has been developed. The resulting alloy is described by 19 parameters. 5 parameters describe the properties of the alloy. 14 parameters describe the composition of the alloy. The parameters are normalized to a range of 0 to 1 before being processed by the neural network. The generator in the generative-adversarial network has 4 input layers. The first input layer receives noise to generate different realistic parameters at the same input values. The second input layer is a mask with descriptions of known and unknown parameters. The number of unknown parameters can be up to 10. The third input layer receives the minimum allowable values of the parameters specified by the user. The fourth layer of the generator receives the maximum allowable values of the parameters. The generator output generates 19 parameters describing the alloy based on the parameters specified at the input. The discriminator checks the validity of the prediction made by the generator. The discriminator has 4 inputs. The first input layer of the discriminator receives the forecast made by the generator. The other 3 inputs come from the generator input layers 2,3,4. The generator-adversarial neural network is able to generate the composition and properties of alloys with a mean absolute error value of 0.082 units relative to the normalized range of test data parameters.

1.4 Summary

This study analyzes the reasons and prospects for the application of artificial intelligence in areas important for the development of society, using two case studies as examples. From the previous paragraph the basic steps for the development of artificial intelligence systems are similar, regardless of the field of development and implementation of the developed systems.

1. Literature review. Analysis of the prerequisites and possible prospects of system implementation.
2. Research proposal. Goals and objectives selection.
3. Database collection.
4. Algorithm development.
5. Comparation and data analysis. The algorithm is tested and compared with analogues.
6. Final algorithm implementation. The algorithm is finalized and implemented in the system.

In the process of research these steps were followed. A detailed description of the steps performed in the study is given in the following sections.

1.5 Organization of the document

This thesis consists of the following sections:

Section 1 – Introduction. This section provides a brief description of the main steps of the research conducted, its causes and results.

Section 2 - Background. This section describes in detail the current state of artificial intelligence research in general and for each of the research cases under consideration. Also, this section describes artificial intelligence algorithms for each case study.

Section 3 - Case study 1. Deep Learning Techniques applied to predict and measure finger movement for the patients with multiple sclerosis. This section focuses on the development of a system for measuring 12 finger joint angles based on camera image and is intended for work within the field of medicine to track the movement and limits of hand mobility in multiple sclerosis. Measuring changes in hand mobility allows the progress of the disease and its treatment process to be monitored. This section discusses approaches to this problem and their specifics, compare results with other approaches.

Section 4 - Case study 2. A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications. This section discusses the possibilities of applying neural networks to predict material parameters, neural network training algorithms, data processing and preparation algorithms, the choice of target metrics to assess the quality results of the developed models, the rationale for the selected approaches to address the emerging issues is given. Evolution of proposed ideas and approaches in different social, medical, biomedical, technological, and scientific areas are presented in subsections in the order of the research conducted.

Section 5 - Conclusion. The final section of this dissertation introduces and discusses the different ideas and conclusions extracted from the final evaluation and results of the presented research work. Also, in this section the specific objectives introduced in section 1.1 will be compared with real obtained results and whether they were successfully met during the process. Future steps will be discussed in this section.

2 BACKGROUND

This section provides an overview of the topics discussed in this dissertation. A detailed literature review of each topic is presented in the corresponding case studies, sections 3 and 4.

2.1 AI algorithms

Every year mankind generates more and more data in social, medical, industrial and other fields. High-performance data analysis algorithms are required to process this data and extract useful information from it. In today's world, machine learning and artificial intelligence algorithms are actively used to analyze data (Bengio et al, 2013; LeCun et al, 2015; Anders et al, 2010). Below are the basic algorithms of machine learning and artificial intelligence.

2.1.1 Machine learning

This subsection describes algorithms for classical machine learning without neural networks. These algorithms use different mathematical tools. Each algorithm is optimized for a separate data processing domain.

Linear regression

Linear regression is a regression model used in statistics of the dependence of one variable y on another or several other variables x with a linear dependence function (Boyd, 2010).

Predictive modeling is primarily concerned with minimizing model error or, in other words, predicting as accurately as possible. Linear regression can be represented as an equation that describes the straight line that most accurately shows the relationship between input variables X and output variables Y. This equation requires finding certain B coefficients for the input variables.

Knowing X, Y values can be calculated, and the goal of linear regression is to find the values of B0, B1…Bk coefficients. An example of visualization of a linear regression for one input value x is shown in the Figure 2.1.

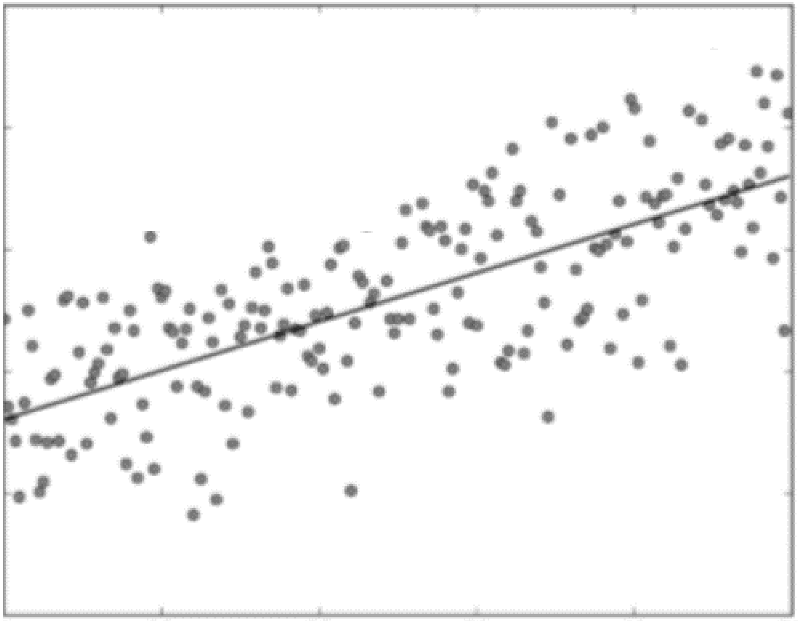


Figure 2.1. Example of the linear regression visualization for one input variable

Various methods like linear algebra or least squares are used to estimate a regression model.

Logistic Regression

Logistic regression is a statistical model used to predict the probability of an event occurring by comparing it to a logistic curve (Boyd, 2010, Martinez et al, 2001). It can be used for binary classification problems.

Logistic regression is similar to linear regression in that it also requires finding coefficient values for input variables. The difference is that the output value is transformed using a non-linear or logistic function.

The logistic function nonlinearly converts any value to a number between 0 and 1. This is very useful because a rule can be applied to the output of the logistic function to bind to 0 and 1 and predict a class.

Because of the way the model is trained, logistic regression predictions can be used to represent the probability that a sample belongs to class 0 or 1. This is useful in cases where you need more justification for a prediction.

Logistic regression performs its task better when redundant and similar variables are removed. The logistic regression model learns quickly and is well suited to binary classification tasks. An example of visualization of a logistic regression is shown in the Figure 2.2.

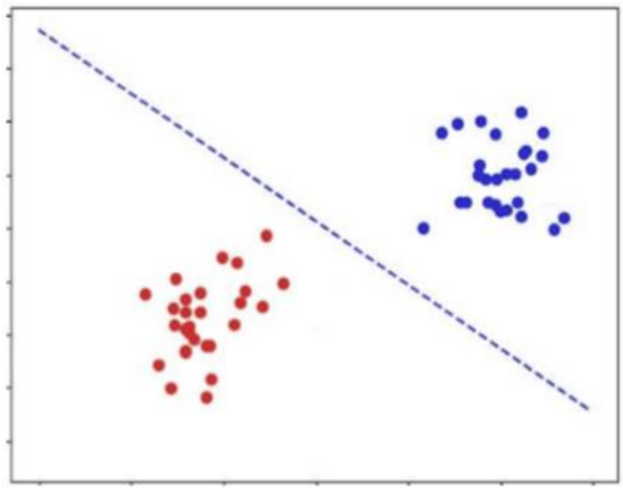


Figure 2.2. Example of the logistic regression visualization

Linear Discriminant Analysis (LDA)

Logistic regression is used when you want to assign a sample to one of two classes. If there are more than two classes, the LDA (Linear discriminant analysis) algorithm can be used.

The data representation for LDA consists of statistical properties of the data calculated for each class. For each input variable, the representation includes the mean value for each class and the variance calculated over all classes (Tin et al, 1998).

Predictions are made by calculating the discriminant value for each class and selecting the class with the highest value. The data is assumed to have a normal distribution, so it is recommended to remove abnormal values from the data before proceeding.

Decision trees

A decision tree can be represented as a binary tree, familiar to many people from algorithms and data structures. Each node represents an input variable and a split point for that variable (Kanungo et al, 2002).

The leaf nodes are the output variable that is used for prediction. Predictions are made by traversing the tree to a leaf node and outputting a class value on that node.

Trees learn quickly and make predictions. They are also accurate for a wide range of tasks and require little data training.

Naive Bayesian classifier

The model consists of two types of probabilities that are calculated using training data. Once the probability model is calculated, it can be used for prediction with new data using Bayes' theorem.

Naive Bayes is called naive because the algorithm assumes that every input variable is independent. This assumption is not always consistent with real data. Nevertheless, this algorithm is very effective for a variety of complex problems like classifying spam or recognizing handwritten numbers.

K-Nearest Neighbor (KNN)

The KNN, K-nearest neighbors, model is represented by the whole set of training data (Chang et al, 2011).

Prediction for a new point is done by searching for K nearest neighbors in the dataset and summing the output variable for these K instances.

KNN may require a lot of memory to store all the data, but it will make the prediction quickly. Also, the training data can be updated so that the predictions remain accurate over time.

The nearest-neighbor idea may not work well with multidimensional data (many input variables), which will negatively affect the efficiency of the algorithm when solving the problem. This is called the curse of dimensionality; it is worth using only the most important variables for prediction.

Vector Quantization Networks (LVQ)

The disadvantage of KNN is that you have to store the entire training dataset. If KNN has performed well, then it makes sense to try the LVQ, Learning vector quantization, algorithm, which lacks this disadvantage.

LVQ is a set of code vectors. They are chosen randomly in the beginning and adapted over a certain number of iterations to best summarize the whole dataset. After training, these vectors can be used for prediction in the same way as in KNN (Chang et al, 2011). The algorithm searches for the nearest neighbor (the best-fitting code vector) by calculating the distance between each code vector and the new data instance. Then a class (or a number in the case of regression) is returned for the best-fitting vector as a prediction. A better result can be achieved if all the data are in the same range, e.g., 0 to 1.

The Support Vector Method (SVM)

The Support Vector Method, one of the most popular and discussed machine learning algorithms (Cortes et al, 1995, Friedman, 2001).

A hyperplane is a line dividing the space of input variables. In the reference vector method, the hyperplane is chosen to best divide points in the input variable plane by their class: 0 or 1. In the two-dimensional plane, this can be represented as a line that completely separates points of all classes. During training, the algorithm looks for coefficients that help better separate the classes with the hyperplane.

The distance between the hyperplane and the closest data points is called the difference. The best or optimal hyperplane separating the two classes is the line with the largest difference. Only these points are relevant in determining the hyperplane and in constructing the classifier. These points are called reference vectors. Special optimization algorithms are used to determine the values of the coefficients that maximize the difference.

Boosting and AdaBoost

Boosting is a family of ensemble algorithms, the essence of which is to create a strong classifier based on several weak ones (Dietterich, 2000, Bengio, 2009). To do this, one model is created first, then another model that tries to correct errors in the first model. Models are added until the training data are perfectly predictable or until the maximum number of models is exceeded.

AdaBoost was the first successful boosting algorithm developed for binary classification. It is the best place to start introduction to boosting. Modern methods like stochastic gradient binning are based on AdaBoost(Bengio, 2009).

AdaBoost is used in conjunction with short decision trees. After the first tree is created, its effectiveness on each training object is tested to see how much attention the next tree should give to all objects. Those data that are hard to predict are given more weight, and those that are easy to predict are given less weight. Models are created one after the other in sequence, and each one updates the weights for the next tree. After all the trees are built, predictions are made for the new data, and the performance of each tree is determined by how accurate it was on the training data.

Because this algorithm places great emphasis on correcting model errors, it is important that there are no anomalies in the data.

2.1.2 Neural networks

This subsection covers machine learning algorithms based on neural networks. 3 main types of neuron layers, fully connected, convolutional and recurrent layers can be selected. Their mathematical structure is similar, the main differences are in the algorithms of their application.

Feed forward neural networks

Feed forward neural networks (FF or FFNN) and perceptrons are straightforward, conveying information from input to output (LeCun et al, 2015; Bengio, 2009; Baek et al, 2020). Each layer consists of input, hidden or output cells. Neurons of one layer are not connected to each other, and neighboring layers are usually fully connected. FFNN is usually trained using a backward error propagation method, in which the network receives sets of input and output data. This process is called supervised learning, learning with a teacher, and it differs from learning without a teacher, unsupervised learning, in that in the latter case, the network composes the output data set independently. The above error is the difference between input and output. If a network has enough hidden neurons, it is theoretically able to simulate the interaction between input and output data. In practice, such networks are rarely used, but they are often combined with other types to produce new ones.

Example of the architecture of the feed forward neural network is shown in the Figure 2.3.

Figure 2.3. Example of the architecture of the feed forward neural network

Convolutional neural networks (CNN)

Convolutional neural networks (CNN) and deep convolutional neural networks (DCNN) are different from other types of networks. They are usually used for image processing, less often for audio. A typical application of CNNs is image classification. Such networks typically use a kernel matrix, checking pixels of the image sequentially rather than checking all the data at once. This input is then passed through convolutional layers in which not all nodes are connected to each other (LeCun et al, 2015; Bengio, 2009; Mallat, 2016; Albawi et al, 2017; Li et al, 2021). These layers tend to compress with depth, often using degrees of two: 32, 16, 8, 4, 2, 1. To the last layer of the CNN is attached FFNN for further data processing.

Example of the architecture of the convolutional neural network is shown in the Figure 2.4.

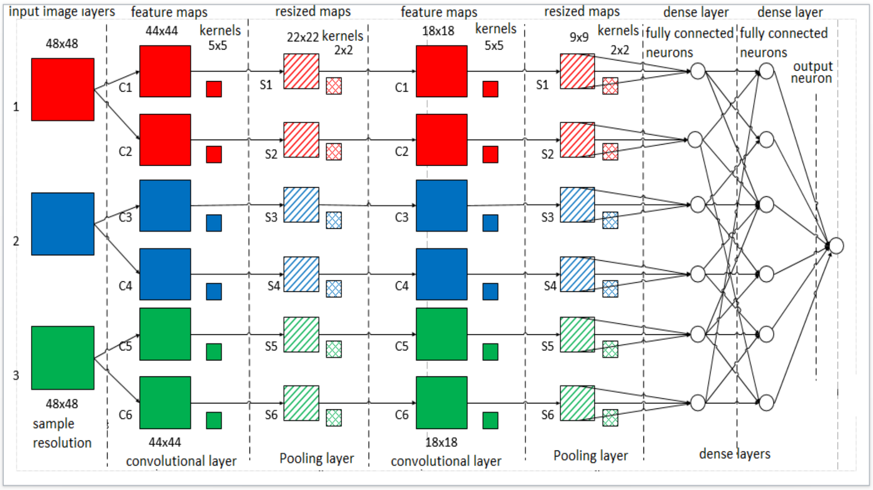


Figure 2.4. Example of the architecture of the convolutional neural network

Generative adversarial networks (GAN).

Such networks consist of any two networks, one of which generates content and the other evaluates it. The discriminator network receives training or generator-generated data. The degree to which the discriminator guesses the data source is further involved in the formation of the error (LeCun et al, 2015; Bengio, 2009; Tzeng et al, 2017). Thus, there is a competition between the generator and the discriminator, where the former learns to generate realistic data, and the latter learns to recognize the generated data. When training generative-adversarial networks, not only do you need to train each of them, but also adjust the balance.

Example of the architecture of the generative adversarial neural network is shown in the Figure 2.5.

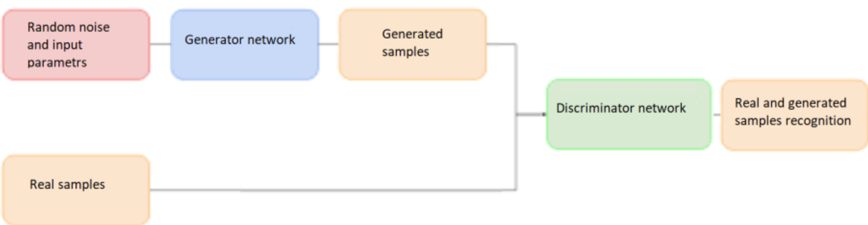


Figure 2.5. Example of the architecture of the generative adversarial neural network

Recurrent neural networks (RNN)

Recurrent neural networks (RNN) are FFNN-type networks, but with a peculiarity: neurons receive information not only from the previous layer, but also from themselves of the previous pass (LeCun et al, 2015; Bengio, 2009; Hochreiter et al, 1997; Yu et al, 2019). This means that the order in which you feed and train the network becomes important. A big complexity of RNNs is the problem of the vanishing (or explosive) gradient, which is the rapid loss of information over time. It only affects the weights, not the states of the neurons, but that is where the information accumulates. Usually networks of this type are used for automatic addition of information

Structure of the recurrent neuron is shown in the Figure 2.6.

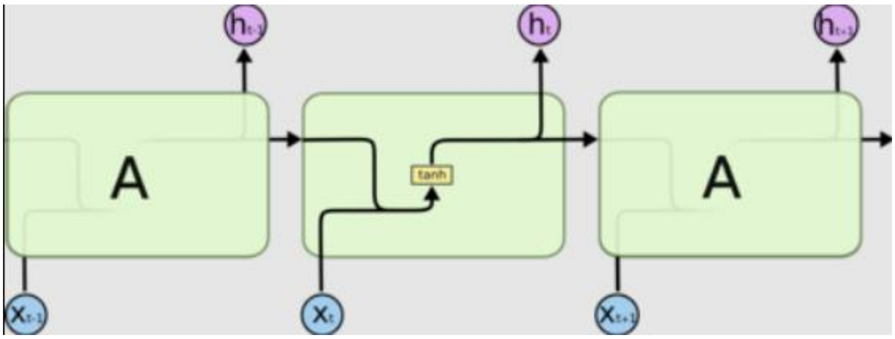


Figure 2.6. Structure of the recurrent neuron

Long short term memory (LSTM)

Long short term memory (LSTM) networks try to solve the aforementioned problem of information loss by using filters and an explicitly defined memory cell. Each neuron has a memory cell and three filters: input, output and forgetting. The purpose of these filters is to protect information. The input filter determines how much information from the previous layer will be stored in the cell. The output filter determines how much information the next layers will receive. A forgetting filter also serves a useful function: for example, if a network is studying a book and goes to a new chapter, some characters from an old one may be forgotten. Such networks can learn to create complex structures, but they consume a lot of resources (LeCun et al, 2015; Bengio, 2009; Hochreiter et al, 1997).

Structure of the long short term memory neuron is shown in the Figure 2.7.

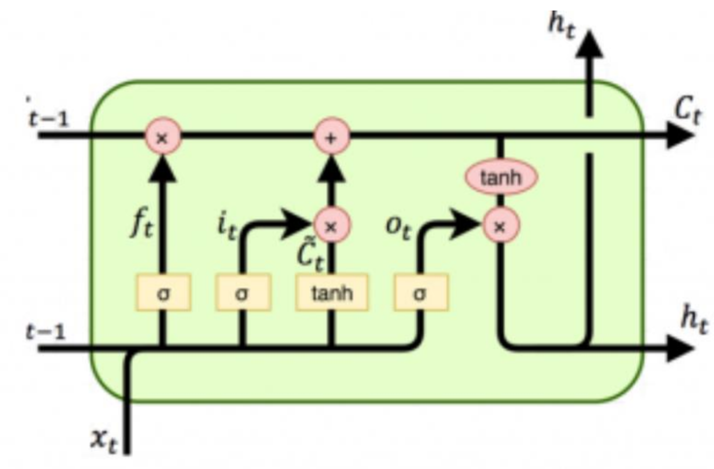


Figure 2.7. Structure of the LSTM neuron

Gated recurrent units (GRU)

Gated recurrent units (GRU) are a small variation of the previous LSTM based network. They have one less filter, and the connections are implemented differently (LeCun et al, 2015; Bengio, 2009; Che et al, 2018). The update filter determines how much information remains from the previous state, and how much is taken from the previous layer. The reset filter works like a forgetting filter.

Structure of the long short term memory neuron is shown in the Figure 2.8.

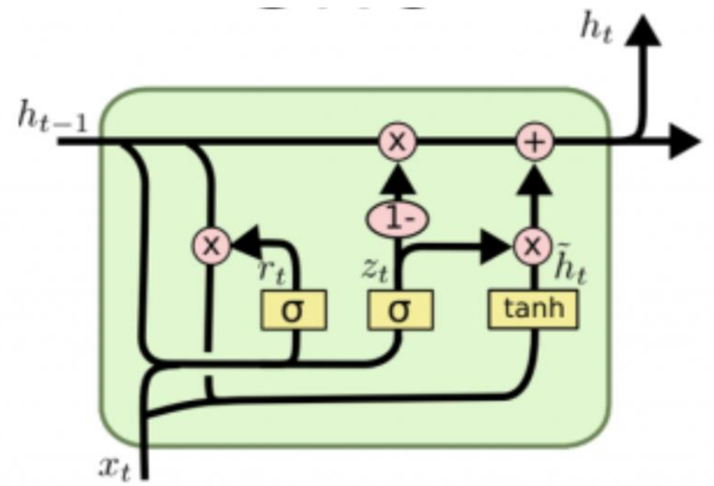


Figure 2.8. Structure of the GRU neuron

Bidirectional RNN, LSTM and GRU

Bidirectional RNN, LSTM and GRU (bidirectional recurrent neural networks, bidirectional long/short term memory networks and bidirectional gated recurrent units, BiRNN, BiLSTM and BiGRU) differ slightly from their unidirectional versions. The difference is that these networks use not only data from the "past," but also from the "future. For example, an ordinary LSTM-type network is trained to guess the word "fish" by feeding the letters one by one, while a bidirectional network is trained to feed the next letter in the sequence as well. Such networks are able, for example, not only to expand the image at the edges, but also to fill the missing data inside(LeCun et al, 2015; Bengio, 2009; Ullah et al, 2018).

Structure of the bidirectional neuron is shown in the Figure 2.9.

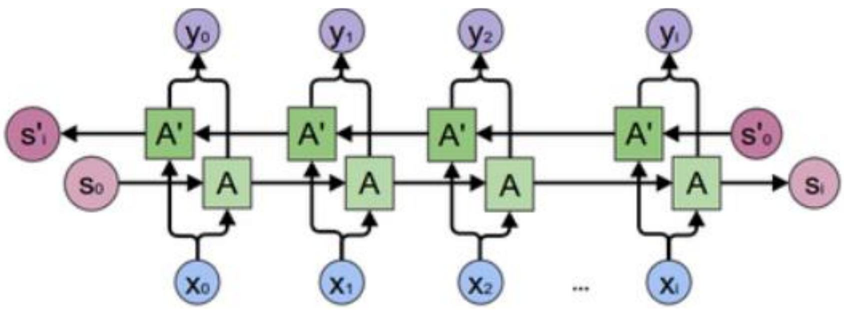


Figure 2.9. Structure of the bidirectional neuron

2.2 Sustainable Development Goals 2030

The Sustainable Development Goals, SDGs, Global Goals, were adopted by the United Nations in 2015 as a universal call to action to end poverty, protect the planet, and ensure that by 2030 all people live in peace and prosperity.

The 17 SDGs are comprehensive-they recognize that actions in one area will affect outcomes in others, and that development must balance social, economic, and environmental sustainability. List of the 17 SDGs: no poverty, zero hunger, good health and well-being, quality education, gender equality, clean water and sanitation, affordable and clean energy, decent work and economic growth, industry, innovation and infrastructure, reduced inequalities, sustainable cities and communities, responsible consumption and production, climate action, life below water, life on land, peace, justice and strong institutions, partnerships for the goals.

Achieving the SDGs in any context requires the creativity, know-how, technology, and financial resources of the entire society.

Described cases of study is related to the 3 and 9 points of the Sustainable Development Goals 2030 plan. Artificial intelligence approaches can help achieve the goals presented in these points of the Sustainable Development Goals 2030 plan.

Case study 1 “Deep Learning Techniques applied to predict and measure finger movement in patients with multiple sclerosis” is related to goal 3 “Good health and well-being” of the Sustainable Development Goals 2030.

Case study 2 “A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications” is related to goal 9 “Industry, innovation and infrastructure” of the Sustainable Development Goals 2030.

2.2.1. Sustainable Development Goals 2030. Good health and well-being

Great progress has been made against several major causes of death and disease. Life expectancy has increased significantly, infant, and maternal mortality rates have fallen, and deaths from malaria have been cut in half.

Good health is essential for sustainable development, and the 2030 Agenda reflects the complexity and interconnectedness of these two challenges. It considers growing economic and social inequalities, rapid urbanization, threats to the climate and environment, the continuing burden of HIV and other communicable diseases, and emerging issues such as noncommunicable diseases. Universal health coverage will be integral to achieving the goals, eradicating poverty and reducing inequality.

Good health and well-being goal has some objectives.

1. By 2030, reduce the global maternal mortality ratio to less than 70 per 100,000 live births

2. By 2030, end preventable neonatal and under-5 mortality, with all countries aiming to reduce neonatal mortality to no less than 12 per 1,000 live births and under-5 mortality to no less than 25 per 1,000 live births.

3. By 2030, end AIDS, tuberculosis, malaria, and neglected tropical diseases, and control hepatitis, waterborne diseases, and other infectious diseases

4. By 2030, reduce premature deaths from non-communicable diseases by one-third through prevention and treatment and promote mental health and well-being.

5. Strengthen prevention and treatment of substance abuse, including substance abuse and harmful use of alcohol

6. By 2030, reduce by half the global number of deaths and injuries from road traffic crashes

7. By 2030, provide universal access to sexual and reproductive health services, including family planning, information and education, and integrate reproductive health into national strategies and programmes

8. Achieve universal health coverage, including financial risk protection, access to quality essential health services and access to safe, effective, quality and affordable essential medicines and vaccines for all.

9. By 2030, significantly reduce the number of deaths and illnesses from hazardous chemicals and air, water, and soil pollution and contamination

10. Strengthen implementation of the World Health Organization Framework Convention on Tobacco Control in all countries, as needed

11. Support research and development of vaccines and medicines for communicable and non-communicable diseases that primarily affect developing countries, provide access to affordable essential medicines and vaccines consistent with the Doha Declaration on the TRIPS Agreement and Public Health, which affirms the right of developing countries to use, to the full, the flexibilities contained in the Trade Related Aspects of Intellectual Property Rights Agreement to protect public health, and in particular

12. Substantially increase health care financing and the recruitment, development, training, and retention of the health workforce in developing countries, particularly in least developed countries and small island developing states

13. Strengthen the capacity of all countries, in particular developing countries, for early warning, risk reduction, and management of national and global health risks

Case study 1 “Deep Learning Techniques applied to predict and measure finger movement in patients with multiple sclerosis” is related to points 4, 8 and 13 in objectives list for the good health and well-being goal of the

Sustainable Development Goals 2030.

First case study explored the potential of neural networks for single camera hand position tracking tasks. The system developed in the study has a high degree of availability. The system can operate in real time, has low cost, is easy to maintain, and is able to track the development of hand mobility limitations without the use of expensive sensors. The accessibility of the system coincides with the provisions described in 8th and 13th points.

The increase in life expectancy entails an increase in the impact of neurodegenerative diseases on a person's life. The importance of treating them and monitoring the progression of neurodegenerative diseases. Impaired mobility due to neurodegenerative diseases can lead to a significant decrease in a person's quality of life. The development of multiple sclerosis can have a significant impact on a person's mobility. In many cases, the symptoms of multiple sclerosis are most active in limbs with many degrees of freedom, in the hands where there are many joints. The degree of course of the disease and the process of its treatment can be judged by the limitations of mobility of the hands. 1st case study can also be referred to 4th point of good health and well-being goal objectives.

Case Study 2, "A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications" is linked to items 8 and 9 on the list for the good health and well-being goal of the Sustainable Development Goals 2030.

This case study explores material analysis using neural networks. Using neural networks for materials analysis tasks can reduce the final cost of materials development, simplify the manufacturing process, and create materials with different target properties. This approach will make widely available materials with different properties, which corresponds to the point 8 of the list for the good health and well-being goal of the Sustainable Development Goals 2030. Also, this approach can help in the development of environmentally friendly materials and means to recycle biology and chemically active substances, which corresponds to the goal 9 of the list for the good health and well-being goal of the Sustainable Development Goals 2030.

2.2.2. Sustainable Development Goals 2030. Industry, innovation and infrastructure

Investment in infrastructure and innovation are critical drivers of economic growth and development. With more than half of the world's population now living in cities, mass transportation and renewable energy, as well as the growth of new industries and information and communication technologies, are becoming increasingly important.

Technological advances are also key to finding long-term solutions to both economic and environmental challenges, such as job creation and energy efficiency. Promoting sustainable industries, investing in research and innovation are all important ways to promote sustainable development.

Industry, innovation and infrastructure goal has some objectives.

1. Develop quality, reliable, sustainable and resilient infrastructure, including regional and cross-border infrastructure, to support economic development and human well-being, with a focus on accessibility and equity for all

2. Promote inclusive and sustainable industrialization and, by 2030, significantly increase the share of industry in employment and gross domestic product in line with national circumstances and double its share in least developed countries

3. Increase the access of small industrial and other enterprises, especially in developing countries, to financial services, including affordable credit, and their integration into value chains and markets.

4. By 2030, modernize infrastructure and retool industry to make it sustainable, with increased resource efficiency and greater adoption of clean and environmentally sound technologies and industrial processes, with all countries acting according to their capabilities

5. Intensify research, improve the technological capabilities of industrial sectors in all countries, especially in developing countries, including, by 2030, encouraging innovation and substantially increasing the number of research and development workers by 1 million, as well as public and private spending on research and development

6. Promote sustainable and resilient infrastructure in developing countries by expanding financial, technological, and technical support to African countries, least developed countries, landlocked developing countries, and small island developing

7. Support the development of domestic technology, research and

innovation in developing countries, including by providing an enabling policy environment for, inter alia, industrial diversification and value addition to commodities

8. Significantly increase access to information and communications technology and strive to provide universal and affordable access to the Internet in the least developed countries by 2030

Case study 2 “A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications” is related to points 4, 5 and 7 in objectives list for the industry, innovation and infrastructure goal of the Sustainable Development Goals 2030.

The application of neural networks to analyze the properties, composition and parameters of materials can allow to automate the process of finding, studying and creating materials with given parameters and structure based on known parameters and from given components.

This approach may allow the development of materials with more pronounced required parameters, solve environmental and economic production by simplifying and reducing the cost of production chains. These arguments allow to realize the point 4 of Industry, innovation and infrastructure goal objectives

Making it easier to develop materials using artificial intelligence techniques will allow more people to develop new materials. The cost of testing new materials will decrease by reducing the cost, complexity, and number of experiments required to achieve the desired material parameters. The emergence of new materials will create new industries and process chains. These arguments allow to realize the points 5 and 7 of Industry, innovation, and infrastructure goal objectives.

2.3 AI and Sustainable Development Goals 2030

Algorithms and artificial intelligence systems are actively developing. Figure 2.10 shows the number of changes in the number of published scientific articles on the topics of artificial intelligence and machine learning.

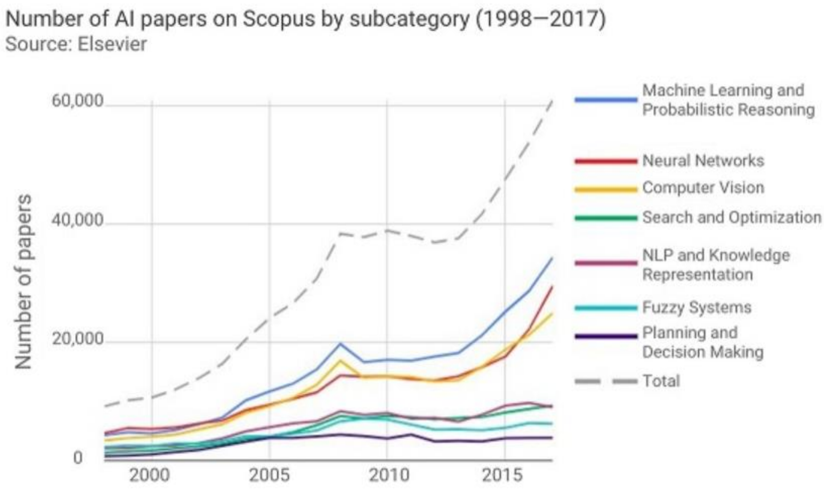


Figure 2.10. Artificial intelligence articles statistic

After 2012 the number of articles began to increase, after 2015 began an exponential growth in the number of articles and active study of artificial intelligence systems based on neural networks. A correlation between the development of neural networks and computer vision algorithms can be observed at this image. This is since the article "ImageNet Classification with Deep Convolutional Neural Networks" demonstrated in 2012 the capabilities and algorithms of parallel machine learning for the task of image classification. This article demonstrated the training capabilities of neural networks on the then available hardware, demonstrated the best results of the proposed algorithm in comparison with classical machine learning algorithms in computer vision tasks.

The active development of machine learning algorithms and neural networks can allow solving different areas of applied problems, including Sustainable Development Goals 2030 goals. The rapid development of neural networks and machine learning algorithms allows them to be applied to a wide range of human activities and to improve performance in areas where neural networks and machine learning are already in use.

This dissertation considers 2 cases of research for two Sustainable Development Goals 2030 plan items.

Case study 1 "Deep Learning Techniques applied to predict and measure finger movement in patients with multiple sclerosis" is related to 4, 8, 13

"Good health and well-being" of the Sustainable Development Goals 2030.

Case study 2 "A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications " is related to points 8, 9 "Good health and well-being" of the Sustainable Development Goals 2030.

Case study 2 "A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications " is related to point 11 "Industry, innovation and infrastructure" of the Sustainable Development Goals 2030.

Currently available algorithms and technologies allow not only to theoretically study and develop algorithms for the cases under study, but also to carry out their practical implementation.

.

3 CASE STUDY 1: DEEP LEARNING TECHNIQUES APPLIED TO PREDICT AND MEASURE FINGER MOVEMENT IN PATIENTS WITH MULTIPLE SCLEROSIS

3.1 Overview

This is case study for medical proposal. This case is focused on the use of computer vision and neural networks to determine the position of the hand. This case study is developed to measure the degree of limitation of hand mobility in the treatment and rehabilitation of people with multiple sclerosis. The algorithm is based on convolutional neural networks.

An algorithm for high-precision recognition of hand parameters in multiple sclerosis and to implement a high-performance algorithm for high-precision recognition of hand position and parameters has been developed. This algorithm is based on the use of neural networks. A neural network model is used for analyzing images coming from one camera. The result of the neural network model analyzing is the measurement of the joint angles of the hand on the image coming from the camera.

This case study focuses on the development of a system for measuring finger joint angles based on camera image and is intended for work within the field of medicine to track the movement and limits of hand mobility in multiple sclerosis. Measuring changes in hand mobility allows the progress of the disease and its treatment process to be monitored.

A static RGB camera without depth vision was used in the system developed, with the system receiving only the image from the camera and no other input data. The case study focuses on the analysis of each image in the video stream independently of other images from that stream, and 12 measured hand parameters were chosen as follows: 3 joint angles for the index finger, 3 joint angles for the middle finger, 3 joint angles for the ring finger, and 3 joint angles for the pinky finger. Convolutional neural networks were used to analyze the information received from the camera, and the case study considers neural networks based on different architectures and their combinations as follows: VGG16, MobileNet, MobileNetV2, InceptionV3, DenseNet, ResNet, and convolutional pose machine.

The final neural network used for image analysis was a modernized neural network based on MobileNetV2, which obtained the best mean absolute error value of 4.757 degrees. Additionally, the mean square error was 67.279 and the root mean square error was 8.202 degrees. This neural network analyzed a single image from the camera without using other sensors.

The resulting neural network developed can measure finger joint angle values for a hand with non-standard parameters and positions

3.2 Introduction

Multiple sclerosis is a potentially disabling disease of the central nervous system, with the symptoms of this disease varying widely and depending on the amount of nerve damage (Cosh et al, 2014). Some people with multiple sclerosis may lose the ability to move, whereas others may experience long periods of remission without any new symptoms.

There is no cure for multiple sclerosis, although treatment can help speed up recovery from attacks, modify the course of the disease, and manage symptoms. Symptoms of multiple sclerosis may differ greatly over the course of the disease depending on the location of the nerve fibers affected. Symptoms often affect movement, such as numbness or weakness in one or more limbs that typically occurs on one side of the body at a time, or the legs, hands or trunk.

Changes in human movement need to be detected to track the disease process and its treatment. People use their hands in all their daily activities, although the structure of the hand is complex, as it has many joints. In the case of multiple sclerosis of the hand, any limitations of each joint must be measured, and development of an automatic system capable of quickly measuring the position of the angles of the joints of the hand will enable the development and treatment of multiple sclerosis to be monitored and treated. Nowadays, these measurements require an expensive set of calibrated medical sensors, and so using only one camera and computer vision algorithms for this task makes such a system cheap and generally available.

There are various methods for assessing the position and parameters of the hand. Many researchers measure this using computer vision techniques, such as gesture recognition, rather than measuring each angle of the finger independently. Many also focus on algorithms to measure other hand parameters besides finger hand position.

An approach based on an application involving computer vision for hand gesture recognition was described in the publications (Chen et al, 2018; Zhou et al, 2016). In these publications separate hand gestures were able to be measured but each angle of the finger could not be measured independently. These approaches cannot be used independently for the angle of the finger.

There are research that focus not only on hand gesture recognition, but also on describing the position of the hand by measuring individual angles of the finger joints. An approach based on strain sensors was described in the work (Kim et al, 2020). In his publication, sensors mounted on the hand measured its joint angle position, obtaining a 64 mean absolute error value of 1.63 degrees.

Publication (Lu et al, 2018) described an approach based on soft strain sensors. In his research, sensors mounted on the hand measure its joint angle position, obtaining an error value of 3.5 degrees.

A non-invasive automatic goniometer test device for joint angle measuring was developed in reference (Tran et al, 2016), where measurement accuracy of the goniometer device proposed was less than 6.6 degrees.

In the reference (Mayer et al, 2017), an approach to detect finger orientation using depth cameras is detailed. This approach does not focus on the medical field of activity and measurement of finger joints, but rather, aims to process the position of the human hand in real time. His research measures the position of the fingers of the hand based on data obtained from the entire 3D shape of the hand, with the Root Mean Square Error, RMSE, error for finger detection position being 8.74 degrees.

A system for estimating the natural postures of a user's fingers from the images was developed in reference (Maruyama et al, 2018). Images were captured by an omnidirectional video camera attached to the center of the user's palm in real time. The positional relationship between the camera and the user's fingers, the length between the finger joints, and interdependencies between the finger joints were provided for such purpose, and the error value of the system developed was five degrees.

A system for estimating the natural postures of a user's fingers from the images captured by a fisheye camera has been developed and described in the publication (Park et al, 2020). This research obtained average errors with an approximate value of 20 mm for fingertip tracking across the different device sizes. The contact finger and hand posture classifiers showed approximately 83% and 90% accuracy, respectively, across the device sizes.

In the publication (Kılıboz et al, 2015), a system for recognizing trajectory-based dynamic hand gestures in real time was developed for human–computer interaction. This research obtained gesture detection and recognition performance to 73% accuracy in a stream of motion.

The publication detailed in reference (Fan et al, 2020) described a real-time on-device hand tracking pipeline that predicts a hand skeleton from a single RGB camera. The mean regression error value normalized by palm size obtained was 13.4%.

An electronic skin integrated with a deep neural network was developed by authors of publication (Kim et al, 2020). The developed system captures dynamic motions from a distance without creating a sensor, and can be used in health-monitoring, motion tracking, and soft robotics.

A neural network-based algorithm with spatial pyramid pooling was proposed in reference (Ashiquzzaman et al, 2020), which decodes gestures or sign language fingerspelling from videos. The algorithm described provided high benchmark accuracy with a fast-processing speed.

In reference (Kim et al, 2017), a hand gesture recognition sensor was developed using ultra-wideband impulse signals. Convolutional neural network was used for reflected waveforms analysis for American sign language gesture classification, with average recognition accuracy also being above 90%.

Authors of reference (Rosenberger et al, 2021) developed an approach for safe and object-independent hu-man-to-robot handovers using real time robotic vision and manipulation based on human body part segmentation and hand/finger segmentation. The resulting robot was able to successfully interact with a human and take the object from the human in 81.9% of trials.

In the work detailed in reference (Woo et al, 2020), data was measured using functional near-infrared spectroscopy during finger tapping tasks. The data augmentation method proposed was used to generate datasets, which in turn. were used to train an AlexNet neural network model. The acquired accuracy attained constituted an improvement compared to that obtained using original data.

The finger vein recognition problem was researched, in the course of which convolutional neural networks for feature extraction for the vein recognition task were used in publication (Chawla et al, 2021). Classification and correct identification accuracy in ranges upwards of 95% and equal error rates below 4% were obtained accordingly.

Many studies focus on specific gesture recognition and hand parameters rather than on determining the position of the hand. These studies also described the above work well when analyzing hands with standard parameters and positions.

This case study focuses on the development of a system for measuring finger joint angles based on camera image and is intended for work within the field of medicine to track the movement and limits of hand mobility in multiple sclerosis. In a healthy hand, the position of the fingers and its joints cannot change independently of each other. However, when the hand is injured, the degree of, finger mobility may change, as people with multiple sclerosis may have a hand with non-standard parameters and positions. As a result, conventional approaches to measuring arm position may not work properly in such cases.

The case study also develops a system capable of measure hand finger joint angles for hands with standard and non-standard parameters and positions. This is possible because training and test datasets of images of the hand are used not only with normal, but also non-standard hand parameters.

Tracking the limitation of hand joint mobility in multiple sclerosis is a complex task, as there are many mobile joints in the hand, the position of which needs to be measured at a specific moment to a good level of accuracy. Measuring changes in hand mobility allows progress of the disease and its treatment process to be monitored, and the use of a single ordinary RGB camera in this case study makes the system developed for finger angle measuring cheaper and easier to install and operate compared to sensor-based systems and specialist cameras. The system developed in this case study will make it possible to take high-precision and high-speed measurements of the finger position at low cost.

Figure 3.2.1 shows the position of the joints on the hand selected.

Figure 3.2.1. The positions of joint angles measured

The 12 measured hand parameters were chosen in this case study as follows: 3 joint angles for the index finger, 3 joint angles for the middle finger, 3 joint angles for the ring finger, and 3 joint angles for the pinky finger.

3.3 Materials and methods

3.3.1 Sample and data

Computer vision and neural network approaches require big datasets. This case study is intended to work within the field of medicine and requires a dataset with hand images and information about hand position in different positions and states.

It is necessary for the dataset to contain not only hand data and images in normal positions and states, but also hands with unusual parameters in unusual positions. Creating a large dataset containing hands with unusual parameters and in unusual positions with use of real data is both difficult and expensive.

A sensor-based hand position measurement approach was considered for the collection of hand position data, whereby a set of sensors is used to detect the position of each finger joint. The dataset obtained from this approach would contains only images of the hands covered by the sensors, and the computer vision system based on a neural network was trained using this dataset. As a result, this system would be unable to work with images without sensors, and so it was decided not to use a sensor-based approach to hand position measurement.

Since the use of sensors was not possible, a simulation approach was considered instead. A program using high polygonal 3D models of hands was developed to create a dataset, with the Unity engine 2019.1 being used to develop the hand simulation program. This program allows to simulate the real hand movement based on 3D model and capture the images of the hand. Pro-gram can change hand textures, colors, proportions, and position of elements of a hand, light and background to be simulated, and also enables the different parameters and positions of the hand to be measured, including the target 12 finger joint angles. Figure 3.3.1 shows a frame from the simulation process with additional markings of joint angles.

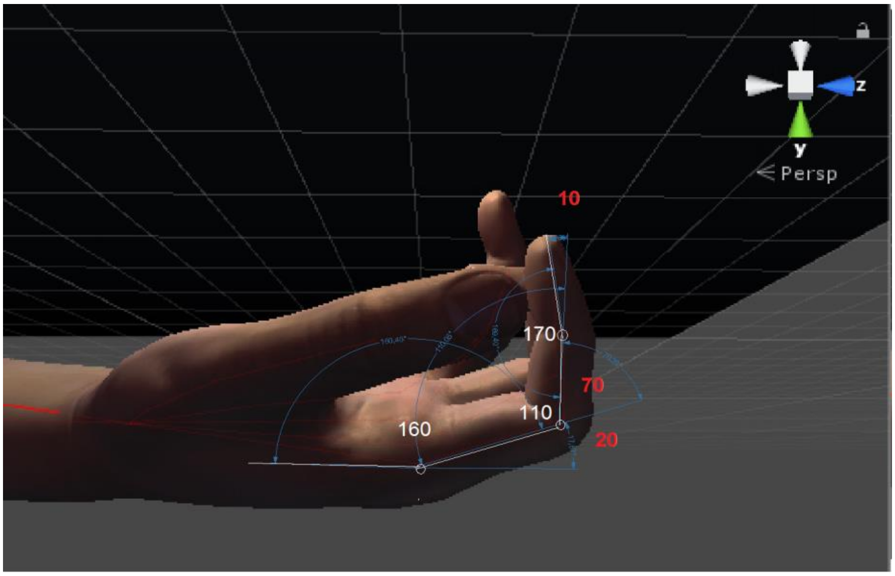


Figure 3.3.1. Frame from the simulation process with additional markings of joint angles.

The hand was rotated 90 degrees for the joint angles visualization. Camera for image capturing for dataset is directed to the front of the hand.

Data regarding the external and internal angles of the joints were collected during the simulation, with the external finger joint angles being used in the calculations by using values for external finger joint angles, it is also possible to calculate the values of the internal corners of the joints.

A dataset was created using this program that contained images of many different hands in different positions with joint angles being measured to a high degree of accuracy for every finger.

The angles of the finger joints on the three axes were changed: X axis values were varied from −10 degrees to 100 degrees, which simulated joint flexion; the Y-axis and Z-axis values were varied from −10 to 10 degrees, which simulated joint displacement. Changes in the values of the finger joint angles were limited so that the texture elements of the hand model did not overlap with each other.

The wrist position of the 3D model of the hand on three axes was also changed as follows: The X axis values were varied from −30 to 45 degrees, which simulated bending the arm at the wrist; on the Y-axis and Z-axis values varied from −15 to 15 degrees, which simulated wrist rotation.

The position of the joints in three-dimensional space relative to the arm model also shifted in the range of −5 to 5 percent relative to the parent model object size.

The size of the individual elements of the hand model was also changed, with the size of each phalanx in the three directions varying from −20 percent to 20 percent of the original size.

The lighting parameters of the 3D model were changed in terms of the intensity of the light, its color, the number of light sources, and their type. Standard Unity engine lighting sources were used for such purpose.

The structure of the 3D hand model was also changed, as was the material from which the hand is made. Each material had its own set of special parameters, although what they had in common were reflectivity values, colors, textures, and normal maps. Changing the parameters of the materials used made it possible to achieve an even greater level of variability, with five different types of materials with 10 different skin textures being used in the study.

Different variants of the background structures and fifty different textures were used, and the structure of the background objects on which these textures were applied was changed. These objects were rotated and moved on three planes, and the hand model was placed at different distances from the background objects.

These changes made it possible to create different hand models at different positions in different environments. In a healthy hand, the position of the fingers and their joints cannot change independently of each other, although the degree of finger mobility may change when the hand is injured. Obtaining hand positions that cannot be achieved by a healthy person makes it possible to expand the applicability of the system described in this study for medical purposes, while other hand parameter recognition algorithms focus on processing hand images without any significant anomalies.

Images from the dataset were used to train the neural network, and the machine learning open-source library TensorFlow and the programming language python 3 were used in the neural network development process. For its part, the programming language OpenCV library was used for additional image processing, such as image noises, filters, blur, and color transformations. This additional image processing allows the influence of random factors on the neural network training, increasing the model robustness.

3.3.2 Models, data analysis and computational methods

Details about the neural networks used in this case study are described in this sub-section.

The architecture of a convolutional neural network was proposed by Yann LeCun (LeCun et al, 1989), with a. convolutional neural network referring to a special type of architecture comprising artificial neural networks aimed at effective image recognition.

The work presented in this case study is based on two neural network architectures: MobileNetV2 (Sandler et al, 2018) and convolutional pose machine (Howard et al, 2017).

Authors of (Sandler et al, 2018) describe the first version of MobileNet, a neural network architecture for mobile and embedded video surveillance systems. This network is based on upgraded architecture that uses depth-separated convolutions to build light and deep neural networks. MobileNet has two simple, global hyperparameters that effectively compensate for delay and accuracy. These hyperparameters allow you to choose the right size model for your application based on the constraints of the task.

MobileNet is based on convolutional blocks, which have 2 layers in this architecture. The first layer is called a depthwise convolution and performs lightweight filtering by applying a single convolutional filter per input channel. The second layer is a 1 by 1 convolution layer with an ReLU activation function, which is responsible for building new features by computing linear combinations of the input channels.

A second version of MobileNet called MobileNetV2 is described in (Howard et al, 2017).

The MobileNetV2 architecture is based on an inverted residual structure, with the residual block input and output providing narrow places. MobileNetV2 uses lighter packages to filter objects at the middle extension level.

MobileNetV2 is based on convolution blocks consisting of two types of blocks. One is a residual block with step 1 and the other is a block with step 2 for size reduction. There are 3 layers for both types of blocks as follows: The first layer is a 1 by 1 convolution with ReLU activation function, whereas the second layer is a depth convolution, and the third layer is a 1 by 1 convolution without any nonlinearity activation function.

Pose Machines provide a sequential prediction framework for learning rich implicit spatial models.

The convolutional pose machine, CPM, is a combination of pose machines and provides a convolutional approach to the neural network. This machine contains framework convolutional layers for learning image features and image-dependent spatial models for pose estimation (Sandler et al, 2018). It implicitly builds model long-range dependencies between variables in structured prediction tasks such as articulated pose estimation.

The convolutional pose machine achieves this by designing sequential architecture comprising convolutional networks that directly operate on belief maps from previous stages and by producing increasingly refined estimates for part locations, with-out the need for explicit graphical model-style inference.

3.3.3 Measurement of variables

Different methods are used to assess the quality of training and neural network performance in the training process. Prediction of the finger joint angle is a regression problem, for which purpose metrics are used as follows: Mean absolute error, mean squared error, and root mean square error.

The mean absolute error, MAE, is used to measure the error between paired observations expressing the same phenomenon. A special feature of MAE is its resistance to emissions in data. Figure 3.3.2. shows the formula for calculation of mean absolute error MAE.

$$MAE = \frac{\sum_{i=1}^{n} |x_i - y_i|}{n}$$

where x - real values
y - predicted values
n- numbers of samples

Figure 3.3.2. Calculation of mean absolute error of MAE.

The mean squared error, MSE, is calculated as being the mean square difference between predicted and actual values. The result is always positive regardless of predicted or actual values, and the ideal value is 0. A square value means that larger errors result in more errors than smaller errors, which in turn means that the model is penalized for larger errors. Figure 3.3.3 shows the formula for calculating the mean square error MSE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2$$

where x - real values
y - predicted values
n- numbers of samples

Figure 3.3.3. Calculation of the mean square error MSE

RMSE is used to measure the difference between the values predicted by the model or evaluator and the values observed. It is calculated as the square root of MSE and large error values have a disproportionate impact on RMSE. Figure 3.3.4 shows the formula for calculating the RMSE.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(x_i - y_i)^2}{n}}$$

where x - real values
y - predicted values
n- numbers of samples

Figure 3.3.4. Calculation of RMSE standard deviation

3.4 Neural network development and training

3.4.1 Adoption of images without preprocessing and marking for convolutional neural network training

The task of measuring joint angles in the image was divided into two subtasks: Measuring the set of parameters of the hand in the image and measuring the finger joint angles.

The subtask of measuring parameters of the hand in the image is algorithmically like the task of classifying images. Algorithms of convolutional neural networks work properly with image analysis and can also work properly with this task, insofar as they are able to create a set of features for images and then send it to analyze and measure the finger joint angles.

The subtask of measuring the finger joint angles by hand parameters is a regression task. In this step, fully connected layers are used to analyze a set of features and measure the finger joint angles.

To measure the parameters of the hand in the image, it was decided to use pretrained convolutional neural networks for the classification of images, and classification layers in these networks were replaced by fully connected layers to work with the regression task. This solution was selected because measuring the finger joint angles by hand parameters is a regression task. A set of features for the hand is created after image processing by the convolutional layers, and fully connected layers allow this set of features to be analyzed and finger joint angles to be calculated.

The following neural network architectures were analyzed: MobileNet, MobileNetV2, InceptionV3, DenseNet121, DenseNet169, ResNet152, ResNet101, and VGG16.

The first approach to training the neural network was taken without preliminary image processing, with images having the following dynamic parameters: Different backgrounds, different lights, different state of hand, different colors, and other different parameters. The mean absolute error, or MAE, for finger joint angle prediction based on this approach was 19–30 degrees, with the best results being obtained by MobileNet, MobileNetV2, and VGG16. Adam optimizer with a learning rate of 0.001 was used in the training process. Batch size comprised 5 images, the number of epochs was 100, and loss function was the mean absolute error.

The best neural network architecture turned out to be MobileNetV2, the mean absolute error value of which was 19.897 degrees. The training process for MobileNet, MobileNetV2, and VGG16 architectures are shown in Figure 3.4.1. It was assumed that in the training process, the neural network cannot identify a good set of features that allow it to measure finger joint angles. For their part, weights of neural networks focus on local features and the training process was stopped at the local minimum.

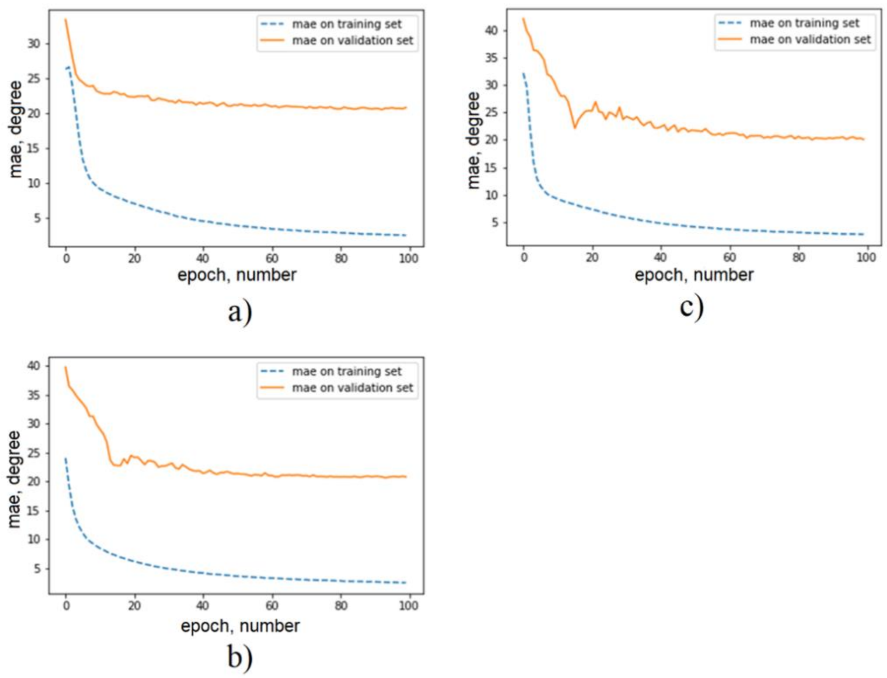


Figure 3.4.1. Mean absolute error values in the training process for neural networks, showing best results: (a) VGG16, (b) MobileNet, and (c) MobileNetV2.

MobileNetV2 produced the best results and was selected for the next stages, although at this stage the approach based on modifying and re-training neural networks for image classification evidenced poor accuracy and major errors.

The approach based on the convolutional pose machine algorithm was considered at the next stage.

3.4.2 Use of convolutional pose machine for measuring joint angles

This approach is based on the use of the open-source project described in (Wei et al, 2016). The image of the hand is processed by a neural network from this project, with the position of the characteristic points of the hand being calculated in this image, and hand position in 2D image space described accordingly.

Use of coordinates of characteristic hand points obtained after image processing by a convolutional pose machine was considered for the purpose of measuring the joint angles. The 2D coordinates of the points were processed using different methods and were sent for processing by a fully connected neural network, this approach providing a result with a MAE between 11–22 degrees. The best result was obtained by transforming the coordinates of characteristic points into information about the line distances between each two points and the angles between each two lines.

This approach also produced results with low accuracy and major errors. It is assumed that in using only the information obtained from coordinates of characteristic points, a lot of information about the hand would be lost. The convolutional pose machine also has its own level of accuracy and error, which also affects the value of the final error. To obtain accurate coordinates of the characteristic points of the hand, the image must be of high quality and resolution, although in this case, performance decreased in geometric progression, as did the growth of the image area. The finger joint angles were measured with a mean absolute error value of 11 degrees by image with a resolution of 1024 by 1024 pixels, and this image was processed using a convolutional pose machine in 0.4–0.5 s by a GPU Nvidia 2080ti. Despite further increase in image resolution, the quality of recognition accuracy of characteristic points of the hand failed to increase accordingly.

The first approach was based on modification and retraining of neural networks for image classification and evidenced a poor level of accuracy and major errors, which was also the case when the convolutional pose machine was used. However, this approach may work with images with dynamic parameters.

An approach based on combination of first and second approaches was also considered.

3.4.3 Adoption of images with marking applied for CNN training

This approach is based on the idea of applying color markers to the hand image. Figure 3.4.2 shows the image marking process.

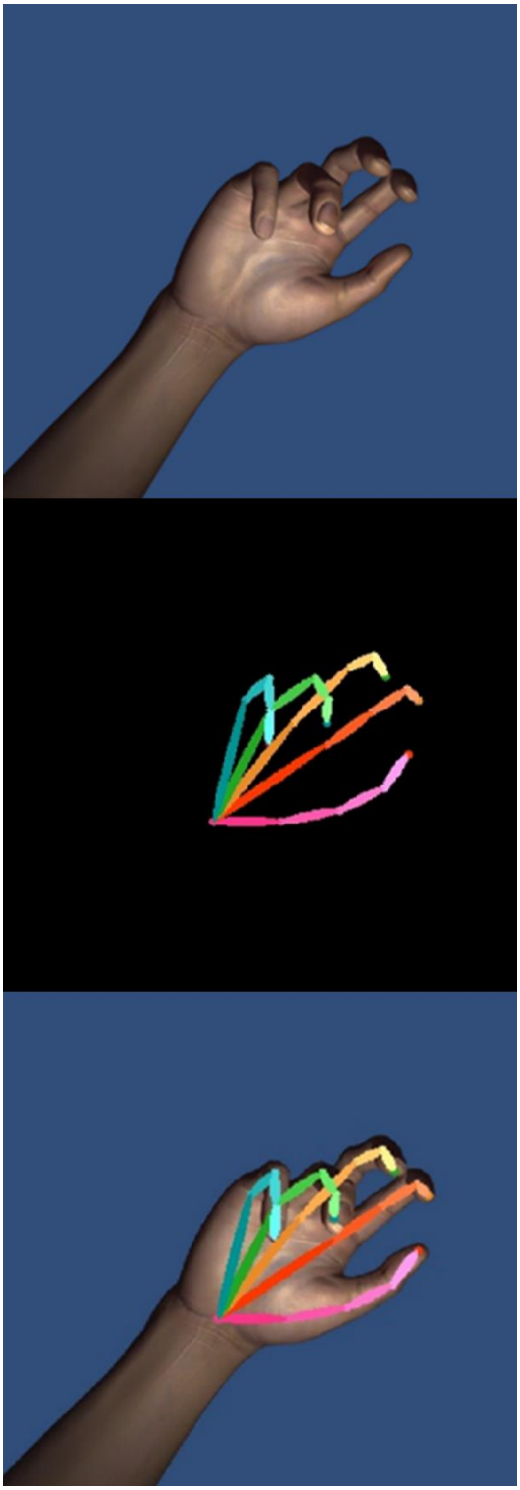


Figure 3.4.2. Image marking process.

The source image on the up side of Figure 3.4.2 is analyzed by convolutional pose machine from previous paragraph and this neural network creates an array with information about characteristic points. The marked image is created from this array, as shown in the center of Figure 3.4.2 In this image, the characterizing points of the hands are marked by different colors and connected to each other by lines, and the combined image on the down side of Figure 3.4.2 is also created. This combined image was used for training purposes.

The convolution kernel of convolutional neural networks reacts most sharply in the presence of sharp color gradients in the image. The markup applied to the image will help in the neural network training process by showing the most important points of the hand, while the markup on the image shows the joint positions for the neural network.

Previously, the convolutional neural network was described as unable to ensure good accuracy for images with dynamical parameters. For optimization of the training process, the marked dataset was based on one constant background and one hand with the moving fingers and static proportions and other parameters. This transformation of the dataset reduces the impact of any factors that may have a negative impact on the training process involved in the neural network. Marked images with one background and static parameters were replaced by original unmarked images and hand images with dynamic parameters at the following stages.

The modified version of the MobileNetV2 neural network produced the best results in the first approach proposed, and so this is the version that was selected. In this case, the last 4 layers of the MobileNetV2 neural network were removed, and the outputs of the “block_16_project_BN” layer were sent to the flattened layer and then analyzed by the fully connected layer of 12 output neurons. Adam optimizer with a learning rate of 0.001 was used in the training process. Batch size comprised 5 images, the number of epochs was 100, and loss function was the mean absolute error.

Different versions of the MobileNetV2 for different image resolutions were tested, with the MobileNetV2 training process for different image resolutions on the marked dataset being shown in Table 3.4.1.

Table 3.4.1. Results of modified MobileNetV2 training for different image resolutions

Width (px)	*Height (px)*	*Mean Absolute Error (Degrees)*	*Root Mean Square Error (Degrees)*	*Number of Model Parameters*
256	256	4.572	8.571	2089036
368	368	3.187	7.817	2396236
440	440	2.687	6.258	2595916
512	512	2.408	6.029	2826316
640	640	2.401	5.911	3379276
768	768	2.398	5.574	4055116

For high accuracy of the neural network with high-resolution images to be ensured, it is necessary to increase the number of network parameters, which slows down the neural network. The network accuracy increases by using high resolution images, however after reaching a resolution of 512 by 512 pixels the accuracy increase slows down sharply. The architecture of the modified MobileNetV2 neural network is shown in Figure 3.4.3.

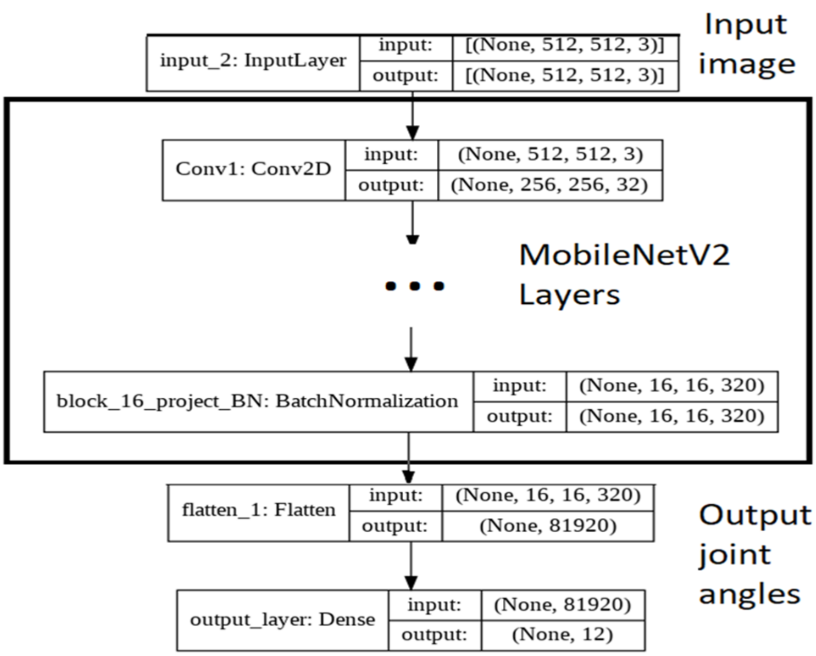


Figure 3.4.3. Architecture of modified MobileNetV2.

Increasing the resolution of the input image above 512 by 512 pixels does not pro-vide any significant increase in the accuracy of the neural network, although it in-creases the size of the neural network and decreases its speed. Based on the data analysis from Table 3.4.1, a resolution of 512 by 512 pixels was selected as the suitable size of the input image.

The training process of modified MobileNetV2 for images of 512 by 512 pixels in size on the marked images is shown in Figure 3.4.4. No data augmentation or dynamic parameters were used at this stage, although marked images of the hand in a fixed position with a constant background were employed. Only the position of the fingers of the hand was changed in the images.

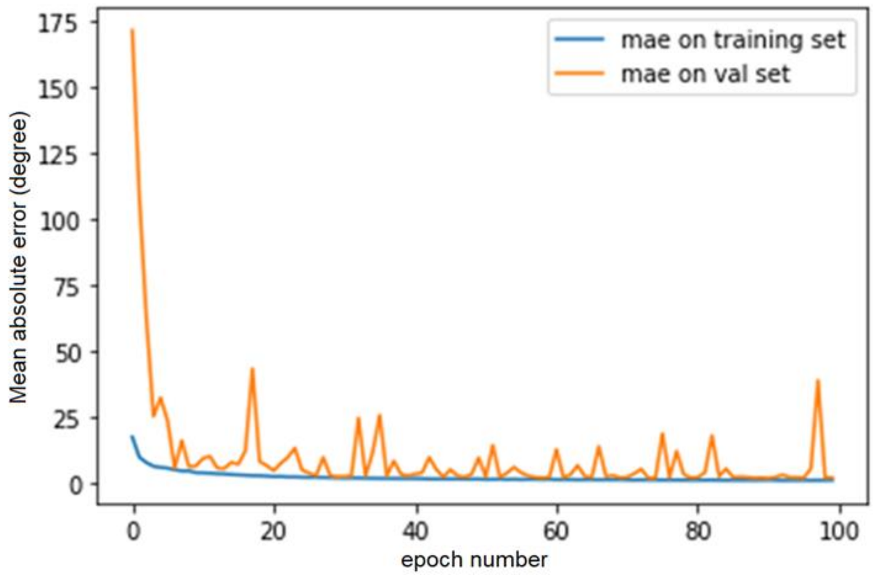


Figure 3.4.4. Mean absolute error values in the neural network training process for marked images of 512 by 512 pixels in size.

As a result of training using the marked images, the best mean absolute error value of 2.408 degrees was obtained, and neural network parameters for this mean absolute error value were saved.

After training using the marked dataset, the marked images in the dataset were then replaced by original images without marking. These images still have no dynamic background hand parameters, but they are not processed by convolutional pose machine.

Pretrained on marked images, the neural network then trains on the unmarked images. Training is provided in 30 epochs with a learning rate of 0.0001, with the loss function being the mean absolute error. The training process is shown in Figure 3.4.5.

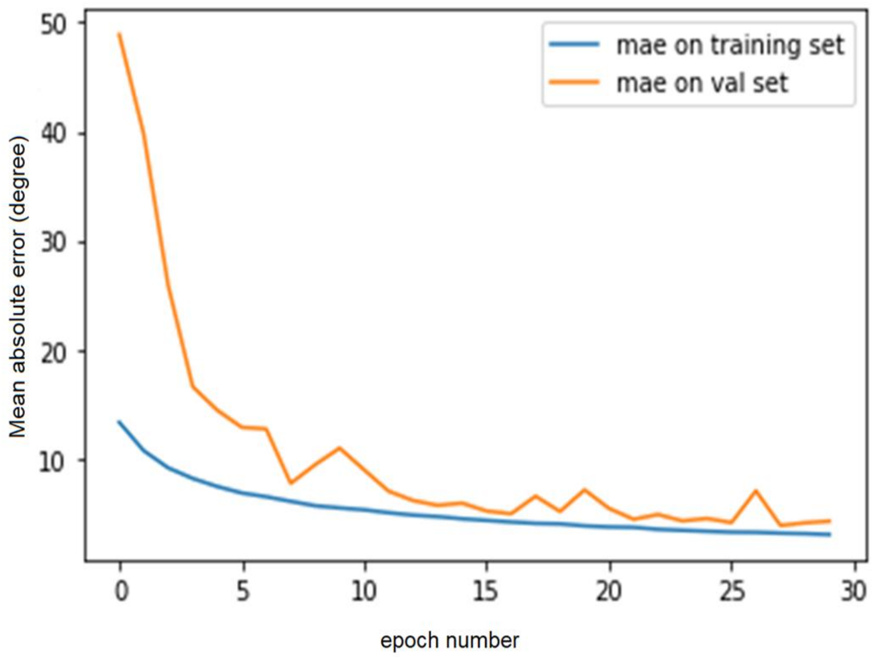


Figure 3.4.5. Mean absolute error values in the neural network training process for unmarked images of 512 by 512 pixels in size.

The neural network was trained on the unmarked images, from which the best mean absolute error value of 2.563 degrees was obtained. The neural network mean absolute error value was increased after training by unmarked images, although performance of the computer vision system had increased significantly. This occurred because images no longer require additional markup obtained using a convolutional pose machine based neural network. The convolutional pose machine was no longer used in the system after this stage.

3.4.4 Use of unmarked images to train the modified MobileNetv2 neural network

After training on the unmarked dataset, some parts of the images in it were replaced by images with dynamic parameters. New hand images have different backgrounds and different hand parameters, and this approach allows the neural network to recognize joint angles in the training process without there being any additional elements or hints in the image. Replaced images were also processed by different filters at this stage as follows: Blur, noise, color filtering, and others. This stage increases the mean absolute error for the dataset, although such processing reduces errors and in-creases reliability of the neural network in real life work.

After pretraining on unmarked images without dynamic parameters, the neural network then trains on the unmarked images with dynamic parameters. Table 3.4.2 shows the accuracy of the neural network after completion of the full training process, depending on the number of images that are replaced at each stage. The minimum mean absolute error value was 4.945, this value being obtained after all unmarked images without dynamic parameters were replaced by images with dynamic hand and back-ground parameters 10 times by parts of 10% size. For each stage we used a learning rate value of 0.001, 40 epochs, and a batch size comprising 5 images. Images were mixed in the training process so as to ensure their uniform distribution in batches used in the neural network training process.

Table 3.4.2 Neural network training results regarding the number of images that are replaced at each stage

Replaced Image Part Per Stage, as a %	*Stages*	*Mean Absolute Error for Final Model, in Degrees*	*Root Mean Square Error for Final Model, in Degrees*
50	2	8.757	10.021
40	3	8.851	10.153
30	4	8.179	10.072
20	5	6.673	9.113
10	10	4.945	8.641
5	20	5.257	8.875

Table 3.4.3 shows the results of the training process with 10% replaced image parts per stage, and also the mean absolute error value at the end of each stage of training. In Table 3 the mean absolute error value increased in the first three stages and then there was a sharp decrease at the following stages. A similar pattern was observed in experiments with a different stage size. The growth of the error at the first stages was because the neural network was originally trained using simplified data and was unable to fully learn to work with unmarked images with dynamic parameters. The neural network recognizes complicated elements of images as noise or error values at the first stages, although all variations of neural networks in the experiment started to reduce the error value on reaching a certain percentage of the image's replacement. Experimentally, it was noticed that error value reduction occurred when 20–30% of images were replaced by versions with dynamical parameters.

Table 3.4.3 Results of training process with 10% replaced image part stage

Step	*Replaced Images, as a %*	*Mean Absolute Error for Final Model, in Degrees*	*Root Mean Square Error for Final Model, in Degrees*
1	*10*	*6.878*	*9.517*
2	*20*	*7.282*	*9.621*
3	*30*	*8.272*	*10.027*
4	*40*	*8.178*	*10.001*
5	*50*	*7.421*	*9.892*
6	*60*	*7.012*	*9.492*
7	*70*	*6.511*	*9.187*
8	*80*	*5.578*	*8.988*
9	*90*	*5.087*	*8.702*
10	*100*	*4.945*	*8.504*

Training process for the last stage of the training process with 10% replaced image part stage shown in Figure 3.4.6.

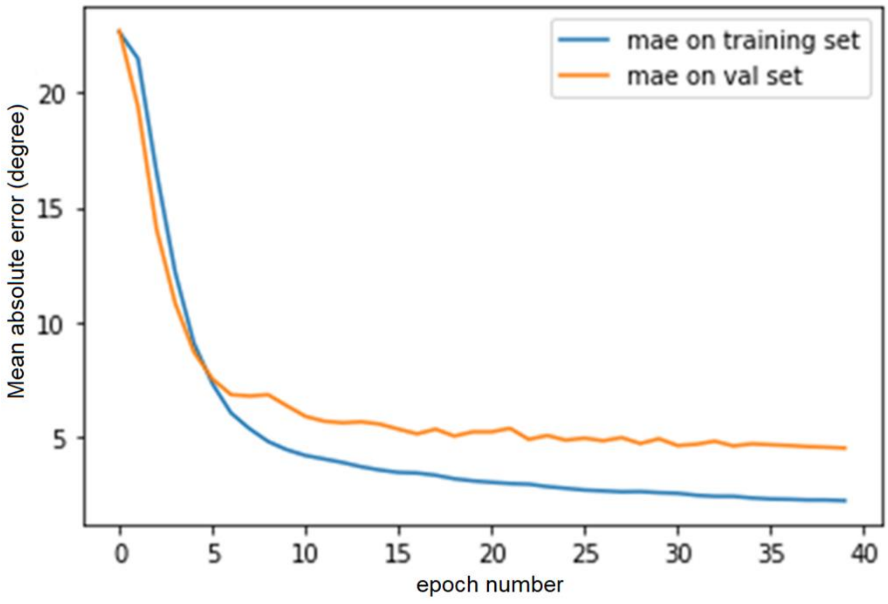


Figure 3.4.6. The last stage of the training process with 10% replaced image part stage.

This neural network in the learning process using undetected images was trained to correctly work with different hand parameters, such as different finger sizes, deformations of the hand, skin color, type of light, variants in the background, image noises, and filters.

3.5 Use of the background subtraction function to improve modified MobileNetV2 neural network results

During training it was noticed that the background of the image had the greatest influence on neural network accuracy. Since the camera is considered stationary, it was decided to use an algorithm to subtract the background, which was implemented using the OpenCV library. Figure 3.5.1 shows the result of background subtraction for the image.

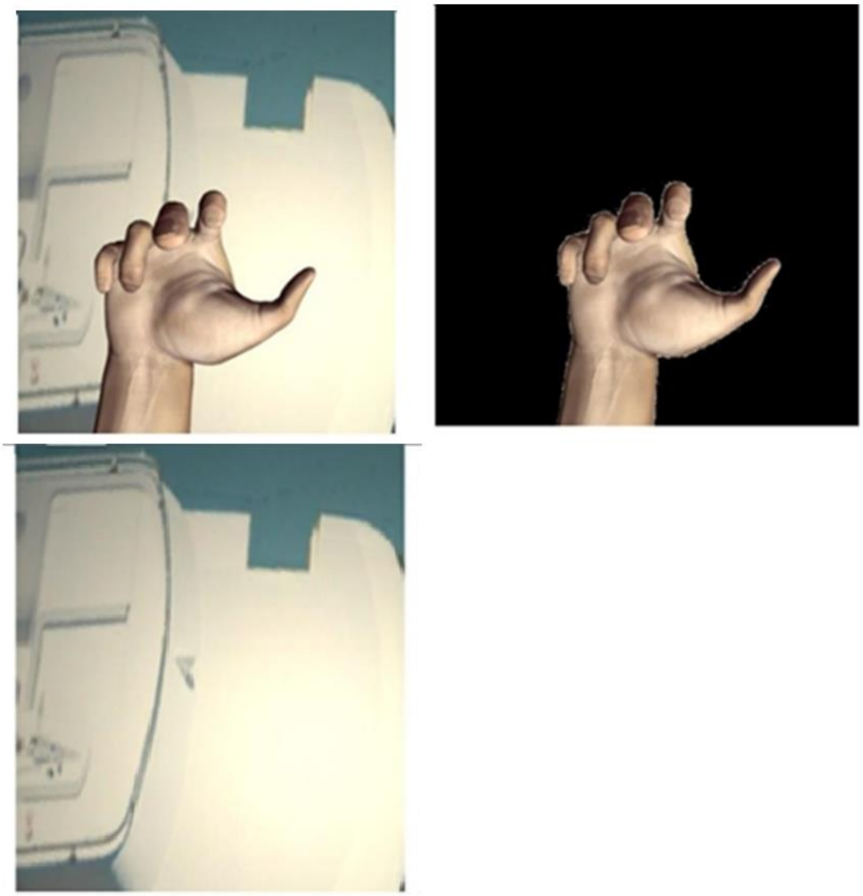


Figure 3.5.1. Result of background subtraction for the image.

After pretraining on images with dynamic parameters, the neural network then trains on the images following background subtraction. These images have different dynamic parameters, resolution, and quality, and are subject to transformation by different filters. Training is provided in 40 epochs with a learning rate of 0.0001, with the loss function being the mean absolute error. The training process is shown in Figure 3.5.2.

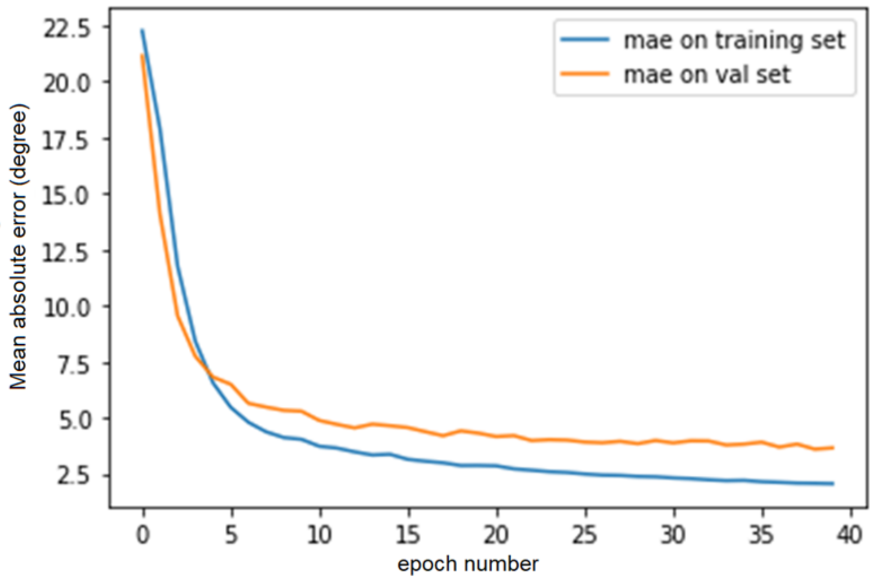


Figure 3.5.2. Mean absolute error values in the training process of neural network for images following background subtraction.

For a neural network trained on images without background, the minimum mean absolute error value was 4.757 degrees, the mean square error was 67.279, and the root mean square error was 8.202.

This neural network analyzes a single image from the camera, and the input image had a resolution of 512 by 512 pixels. This image was processed by the neural network in 7–15 milliseconds using a GPU Nvidia 2080ti.

To train a neural network to find hidden data dependencies, a learning algorithm based on the Deep Dreamer (Mordvintsev et al., 2015) algorithm and Inceptionv3 (Szegedy et al., 2015) neural network was developed. This developed algorithm analyzes the image by a trained neural network. Neural network detect features on the image. Algorithm hides main features on the image. Importance and place of the features is estimated based on neural network layers output values. The new neural network trained on the processed data with hidden features are forced to search and train a new set of features in the data, since the known features have been hidden. Trained on original and processed data neural networks are used in an ensemble. Source image and processed image with hidden features is shown in Figure 3.5.3

Figure 3.5.3. Source and processed by modified DeepDream algorithm images

Inceptionv3 model has been trained for images augmentation. Joint positions on the image 3.5.3 has been edited significantly and has hidden. When second neural network will train on the processed images, it will not be able to detect standard features.

Second MobileNetV2 neural network has mean absolute error value 12.345 after training on the processed image. This neural network is useless without first neural network trained before. But this neural network trained for different features and can be combined with first neural network.

In this case study, two MobileNetV2 networks based neural network models are combined into an ensemble. Combination of these models improves loss function and metric values because each model has been trained can detect different feature sets. Obtained mean absolute error value is 4.702. But performance was decreased.

This ensemble of the MobileNetV2 neural networks analyzes a single image from the camera, and the input image had a resolution of 512 by 512 pixels. This image was processed by the neural network in 20–55 milliseconds using a GPU Nvidia 2080ti. This neural network can not work in real-time mode. But DeepDream based approach can be used in other areas, where real-time work is not important.

3.6 Results

This case study considered an approach to measuring the finger joint angles in fields of medicine based on computer vision and neural networks. The convolution neural network for analysis of static RGB images from the camera for 12 finger joint angle measuring was duly developed for such a purpose. This neural network receives only the images from the camera and does not use any other input data. The case study focused the analysis of each image on the video stream independently of other images from that stream.

Reconstructing a full 3D model of the hand is not necessary in order to achieve the main aim of the this case study. Rather, the aim was to measure the finger joint angles of the hand. This is a usual regression task, hence metrics such as the mean absolute error and mean square error.

A comparison of neural network accuracy with the results obtained from previous research is shown in Table 3.5.1.

Table 3.5.1 Comparison of research results

Algorithm Source	***Main Device***	***Feature***	***Mean Absolute Error, in Degrees***	***Root Mean Square Error, in Degrees***
(Kim et al, 2020)	*Fiber Bragg grating strain sensor*	*Only sensors are used*	*1.63*	-
(Lu et al, 2018)	*Soft strain sensors*	*Only sensors are used*	*3.5*	-
(Tran et al, 2016)	*Goniometer device*	*Use of non-invasive automatic goniometer test device*	*6.6*	-
(Mayer et al, 2017)	*Depth cameras*	*Not aimed at medical use. Use of depth cameras*	-	*8.74*
(Maruyama et al, 2018)	*Omnidirectional camera*	*Use of hand preset information. Use of omnidirectional camera*	*5*	-
Author's proposal	*RGB camera*	*Use of single image from RGB camera. No hand preset information used. Hand image must have resolution of at least 512 by 512 pixels*	*4.945*	*8.504*

Author's proposal	*RGB camera*	*Use of single image from RGB camera. No hand preset information used. Hand image must have resolution of at least 512 by 512 pixels. Subtraction of hand image background applied*	*4.757*	*8.202*
Author's proposal	*RGB camera*	*Use of single image from RGB camera. No hand preset information used. Hand image must have resolution of at least 512 by 512 pixels. Deep Dream based augmentation has been used.*	*4.702*	*7.982*

The results obtained from the neural network cannot be properly compared to those from previous research (Park et al, 20; Fan et al, 20). Research (Park et al, 20; Fan et al, 20) focused on creation of a 3D hand model, measured by other metrics, and it should be noted that such research does not work in the case of a hand with abnormal parameters.

The results obtained from the neural network cannot be properly compared to those from previous research (Kılıboz et al, 2015; Ashiquzzaman et al, 2020; Kim et al, 2017), as this research focused on hand gesture recognition. Gesture recognition is a classification task, and one gesture may represent different positions of the finger joints of the hand, and the task considered in this case study is a regression task for predicting finger joint positions. It is thus not possible to properly compare the classification task and the regression task.

The results obtained from the neural network cannot be properly compared to those from previous research (Kim et al, 2020). The research (Kim et al, 2020) describes a sensor-based system, albeit one focused on describing the parameters of the physical principles of operation by developing a neural network to work with the sensor, although the research [Kim et al, 2020] does not provide results for measuring hand position accuracy. Thus, it is not possible to properly compare properly (Kim et al, 2020) and the system developed in this case study.

The results obtained from the neural network cannot be properly compared to those from previous research (Rosenberger et al, 2021; Woo et al, 2020; Chawla et al, 2021).

The research (Rosenberger et al, 2021) focuses on finding an object in the image, segmenting it, and analyzing its shape to plan further interactions, but does not determine the position of the hand in the familiar human form. The hand is described by the neural network in terms of special features regarding the human-interaction task, and so a comparison of results is not possible.

The research (Woo et al, 2020) focused on finger tapping analyze, but does not describe other hand parameters besides finger tapping. Thus, a comparison of results is not possible.

The research (Chawla et al, 2021) focused on vein recognition analyze, but does not describe other hand parameters or position besides vein recognition. Thus, a comparison of results is not possible.

The best results for the real-time work were obtained after applying a model based on processing images with subtracted background of the hand. These images had different dynamic parameters, resolution and quality, and were subject to transformation by different filters. Specifically, this neural network had a mean absolute error value of 4.757 degrees, with a mean square error of 67.279, and analyzed a single image from the camera. The image had a resolution of 512 by 512 pixels, with the input image being processed by the neural network in 7–15 milliseconds using a GPU Nvidia 2080ti.

This result was obtained via training using a changing dataset, and this training of the neural network was carried out at several stages with consecutive complication of data for training purposes as follows: Training on images with markup and invariable hand parameters in the image, training on images without markup with invariable hand parameters in the image, training on images without markup with dynamic hand parameters in the image, and training on images without markup with dynamic hand parameters in the image after subtracting the background of the image.

The neural network developed was trained on simulated data subjected to strong processing. The use of simulated data made it possible to collect a dataset of images of people's hands with various parameters and in various positions, which is an extremely difficult task in the real world. The application of additive data processing will allow testing with patients in the future since the neural network developed was designed for medical tasks for patients with multiple sclerosis. People may have hands with non-standard parameters and in non-standard positions, and a feature of the neural network is the ability to measure finger joint angle values for hands with non-standard parameters and positions.

3.7 Case study 1. Deep Learning Techniques applied to predict and measure finger movement in patients with multiple sclerosis. Conclusions

This case study focused on the development of a system for measuring finger joint angles based on camera image and was intended to work within the field of medicine to track the movement and limits of hand mobility in multiple sclerosis.

The convolution neural network for analysis of static RGB images from the camera for 12 finger joint angle measuring was developed accordingly, with the best results being obtained after applying a neural network model based on processing images with subtracted background of the hand.

The neural network developed allows non-contact measurement of finger joint angles to be undertaken using a single RGB camera without the use of additional sensors. The use of a single ordinary RGB camera in this case study makes the system developed for finger angle measuring cheaper, and easier to install and operate com-pared to sensor-based systems and specialist cameras. A camera must support at least 30 frames/sec and the resolution 720p for the correct neural network work. A camera with better parameters will give better results.

Image analysis approaches based on different neural network architectures were considered to achieve a lightweight and fast neural network for real-time work.

The option of building a neural network based on convolutional pose machine architecture was considered. However, convolutional pose machine-based neural network architecture is a slow algorithm and not suitable for real-time operation on low computing power devices. It also does not work correctly for hands with unusual parameters. The convolutional pose machine-based neural network architecture was removed from the final version of the neural network to improve performance. However, the results of convolutional pose machine-based image analysis and markup were used to train a light MobileNetV2 based neural network.

The use of MobileNetV2 architecture, originally developed for image neural network processing on mobile devices, in turn makes it possible to run the system on systems with low computing power. The system developed in this case study will thus make it possible to take high-precision and high-speed measurements of the finger position at low cost.

The system can operate in real time, has low cost, is easy to service, and is able to monitor the development of hand mobility limitations without the use of expensive sensors. The high availability of the system is consistent with items 8 and 13 of the "Good health and well-being" section of the Sustainable Development Goals 2030 list.

Early diagnosis of multiple sclerosis, rapid response to the manifestation of its symptoms and tracking the process of treatment can significantly improve the quality of life and life expectancy of people with this disease. The use of the developed system can help to achieve the item 4 of the "Good health and well-being" section of the Sustainable Development Goals 2030 list.

4 CASE STUDY 2: A DEEP LEARNING APPROACH TO PREDICT THE PROPERTIES AND COMPOSITION OF MATERIALS WITH POTENTIAL CAPABILITIES IN BIOMEDICAL APPLICATIONS

4.1 Overview

This case study discusses the possibilities of applying neural networks to predict material parameters, neural network training algorithms, data processing and preparation algorithms, the choice of target metrics to assess the quality results of the developed models, the rationale for the selected approaches to address the emerging issues is given.

The development of new devices and medicine implies the development of an algorithm and a device that implements it in the physical world. Concentrating on using and developing only algorithms without developing new materials significantly reduce the range of problems available for solving. The development of devices with given parameters requires materials with different parameters, properties and compositions.

The results, ideas and explore process were improved iteratively. Evolution of these ideas and approaches are presented in subsections in the order of the research conducted. Prediction of material parameters using neural networks can accelerate and simplify the development of new materials for sustainable social, medical, technological and scientific development.

The following points can be highlighted in the research process:

1. Neural network-based prediction of the critical superconductivity temperature for superconducting materials based on their chemical formula.

2. Neural network-based prediction of the reduced glass transition temperature of metal alloys based on a neural network.

3 Neural network-based prediction of the composition of paper-like materials based on the specified chemical and physical parameters.

4. Neural network for generation of composition and parameters of metal alloys based on specified range of known and unknown parameters.

In these subsections, the issues of predicting the composition and properties of materials are considered from different angles with different approaches.

4.2 Neural network-based prediction of the critical superconductivity temperature for superconducting materials based on their chemical formula.

This subsection discusses the architecture of neural networks and their ensembles for predicting the critical superconductivity temperature of materials based on their chemical formula. The subsection describes the methods and process for extracting data from the chemical formula and preparing this extracted data for use in training neural networks using TensorFlow. Recurrent neural networks including long-term short-term memory layers and neural networks based on one-dimensional convolutional layers are used for data analysis. The proposed model is an ensemble of pre-trained neural network architectures for predicting the critical temperature of superconductors based on their chemical formula. The architecture of the seven pre-trained neural networks is based on long-term short-term memory layers and convolution layers. The final ensemble uses six neural networks: one LSTM-based network and four convolutional neural networks, and one ensemble of convolutional neural networks. The LSTM neural network and convolutional neural network were trained over 300 epochs. An ensemble of models was trained in 20 epochs. All neural networks are trained in two phases. In both phases Adam optimizer was used. The first stage was trained using Mean Absolute Error loss function with learning rate of the optimizer equal to 0.001. The second stage was trained with Mean Squared Error loss function with learning rate equal to 0.0001. The final ensemble is trained with a learning rate equal to 0.00001. The final ensemble model has the following accuracy values: Mean Absolute Error is 4.068, Mean Squared Error is 67.272, and Coefficient of Determination is 0.923. The final model can predict the critical temperature for the chemical formula with an accuracy of 4.068 degrees

4.2.1. Introduction

This study presents work on superconducting materials, materials that conduct current with zero resistance at or below the critical temperature Tc (Hamidieh, 2018). Most of the known superconductors exhibit the effect of superconductivity at extremely low temperatures below 100K. However, despite the need for low temperatures to exhibit the effect of superconductivity, superconductors are used in many fields. Although the superconductivity effect can be used in many applications, it disappears when the temperature rises above the critical temperature. The need to maintain low temperatures to maintain the effect of superconductivity is a difficult and costly task

The relevance of superconductors includes societal challenges related to health and wellbeing. Superconductors are used in medicine mainly inside devices for CT scan and Magnetic Resonance Imaging, MRI, systems and in magnetometers for SQUID, Superconducting Quantum Interference Device (Hamidieh, 2018). They are used for magnetoencephalography, magnetocardiography and other processes for detection and mapping weak magnetic fields of the human body. Introduction of superconductors in such systems allows to conduct safe for the patient's research methods, to obtain a highly accurate three-dimensional picture of the state of the studied area of the human body.

Although the superconductivity effect can be used in many areas, the effect disappears when the temperature rises above critical. The need to maintain low temperatures to maintain the superconductivity effect is a complex and costly task. So, increasing the temperature of MRI (Noe et al, 2007) coils of the device above the critical temperature destroys the effect of superconductivity and triggers the process of destruction of the coils. This effect is used for emergency stop of MRI machines. The device is out of service and requires a long and expensive repair after emergency stop. However, the use of high-temperature superconductors in the superconducting MRI coils would avoid such results of an emergency shutdown, making this procedure safe.

In addition, superconductors are often used in electrical, researchers and other systems. Superconductors are used in superconducting fault current limiters, SCFCKs, for electrical current limitation (Hamidieh, 2018). Superconducting coils used for generation and sustain high magnetic fields in the Large Hadron Collider at CERN (Hamidieh, 2018). Superconductors are used to create and support quantum states in quantum systems and quantum computers (Noe et al, 2007). Superconductivity effect is used in the development of electric motors, unipolar machines, topological generators, rigid and flexible cables, switching and current limiting devices, and magnetic separators, transport systems, in the production of such coils for accelerators and for creation of devices for measuring temperatures, costs, levels, and pressures.

Superconductors are often used in electrical, research, and other systems. Superconductors are used in superconducting fault current limiters, SCFCK, to limit electrical current (Noe et al, 2007). Superconducting coils are used to create and maintain high magnetic fields in the Large Hadron Collider at CERN (Hamidieh, 2018). Superconductors are used to create and maintain quantum states in quantum systems and quantum computers (Krinner et al, 2019). The effect of superconductivity is used in the development of electric motors, unipolar machines, topological generators, rigid and flexible cables, switching and current-limiting devices, as well as magnetic separators, transport systems, in the production of such coils for gas pedals and to create devices for measuring temperatures, flow, levels and pressures.

Currently, there are two main directions in the field of superconductivity applications: in magnetic systems and in electrical machines. There is a lot of research in search of new superconductors with high critical temperatures (Si et al, 2016; Flores-Livas et al, 2016; Sleight et al, 1893).

In the study "A Data-Driven Statistical Model for Predicting the Critical Temperature of a Superconductor" by Kam Hamidieh (Hamidieh, 2018), models were developed to predict the properties of chemical compounds based on XGBoost and multiple regression statistical algorithms. The algorithms were evaluated using RMSE and R2 metrics. Superconductor data were taken from the superconducting materials database maintained by the National Institute of Materials Science of Japan at http: //supercon.nims.go.jp/index_en.html. After some preprocessing of the data, 21263 superconductors are used. The model developed in the study based on the multiple regression method has an RMSE value of 17.6K and R2 of 0.74. The model based on the XGBoost method has an RMSE of 9.5K and an R2 of 0.92.

In the study "A Predicting the Critical Temperature of Superconductors with XGBoost" by Abdulkadir Karacı and Kemal Akyol (Abdulkadir et al, 2019), the model to predict the properties of chemical compounds is based on the statistical algorithm XGBoost. In his paper, he uses a statistical model to predict the critical temperature of a superconductor based on features extracted from the chemical formula of the superconductor. This paper (Abdulkadir et al, 2019), as well as the article by Kam Hamidie (Hamidieh, 2018), is based on the idea of analyzing the features of the chemical formula of the material and develops his ideas. The algorithms were evaluated using the RMSE and R2 metrics. The XGBoost model developed has an RMSE value of 9.091K and R2 of 0.928.

In the study "Critical Temperature Prediction of Superconductors Based on Atomic Vectors and Deep Learning" by Shaobo Li (Shaobo et al, 2019), 2020, a hybrid neural network combining convolutional neural network and neural network with long-term short-term memory is proposed to extract material characteristics for critical temperature prediction of superconductors. The superconductor data are taken from the superconducting materials database maintained by the National Institute of Materials Science of Japan. The algorithms were evaluated using RMSE, MAE, and R2 metrics. The developed hybrid neural network has an RMSE value of 83.565Kand R2 of 0.899 and MAE of 5.023.

The above studies use the same data set. Therefore, this subsection presents a new technique to accurately predict these cases.

The purpose of the study is to develop a model to use the chemical formula of the material and then predict the critical superconductivity temperature for that material. This study reviews and describes an approach based on the use of various neural network architectures and combinations thereof to analyze chemical formulas. This study considers the use of neural networks whose structure is based on the use of LSTM and convolutional layers.

4.2.2. Materials

The study used a data set from the study "A Data-Driven Statistical Model for Predicting the Critical Temperature of a Superconductor." This data set used in the follow-up study contains 21263 samples and is publicly available at the University of California, Irvine (Center for Machine Learning and Intelligent Systems). This set contains superconductor formulas and their parameters. The parameters of the superconductor formulas were presented in a table with 21263 rows by the number of superconductor formulas in the database. The formulas in the dataset contain from one to nine chemical elements. The dataset has 87 columns, with columns 1 through 86 describing the chemical elements used in the formulas, and column 87 contains the critical temperature Tc. The table has a coefficient value for each element in each formula. The values of the properties of the elements were obtained from the periodic table of elements in csv format (Periodic Table of Elements).

Elements with atomic number up to 86, elements up to and including radon, were used in the data set. For each element we chose 16 parameters: atomic mass, number of neutrons, number of protons, period, atomic radius, electronegativity, first ionization, density, melting point, boiling point, number of shells, group, specific heat, is a metal, is nonmetal, is metalloid. The parameters "is metal," "is nonmetal," and "is metalloid" were presented in a hot encoding, one hot encoding format. These parameters were chosen because they provide a precise description for each of the 86 elements in question. Once these parameters were chosen, the table of 86 items was standardized for these parameters.

Machine learning algorithms have low data accuracy with large differences in the size of the input values. The chemical element parameters can vary by many orders of magnitude, so it was necessary to adapt the data for a qualitative learning process. For each of the 16 element parameters, a mean value and a mean deviation were calculated based on the data for the 86 elements used. Based on this data, the periodic items table was standardized for better compatibility with machine learning algorithms.

The elements from the standardized chemical elements table were inserted into the superconductor formulas from the element dataset, respectively. The coefficients of the chemical elements in the standardized formulas were not exposed. If the number of elements in the formula is less than 10, then the formula was expanded to 10 and all parameters of all missing elements were set to zero. The obtained result was entered into the processed data array.

We considered two options for the arrangement of the chemical elements in the processed data set: sorting by their indication in the chemical formula in the original data set and sorting by their number in the periodic table of elements. The choice of sorting by the number in the periodic table of elements was due to some differences in the order of the chemical elements in the formulas in different areas of activity. In this variant of sorting the arrangement of elements in the formula representation for neural network the peculiarities of recording the chemical formula do not affect the result of the neural network.

4.2.3. Methods

This subsection details the neural networks used in this study.

LSTM neural networks, neural networks with long-term memory, are a special kind of recurrent neural networks capable of learning long-term dependencies. They were introduced by Sepp Hochreiter and Jürgen Schmidhuber (Understanding LSTM Networks). LSTM-networks are specifically designed to avoid the problem of long-term dependence. LSTM networks consist of LSTM neurons.

LSTM predictions are always based on the past input experience of the network. However, as the size of the input data increases, the importance of the data coming into the neural network at the beginning decreases compared to the data coming into the neural network later. Therefore, the data at the beginning of the sequence has a minimal effect on the result of the LSTM layer of the neural network, and the most recent data has a maximum effect on the result of the LSTM layer. To reduce the effect of this effect, the dimensionality of the data is added to the values used for the neural network. In this study, a value of 0 is used to augment the data to the standard size used by the neural network. These fills are added before the data is processed so that adding the data to the standard size accepted by the LSTM layer for processing will have minimal impact on information extraction from the data. If these fills are added after the data is processed, the result of the neural network will be less accurate because the values of the fills will have a significant impact on the result.

Deep convolutional networks provide state-of-the-art classification and regression results for many high-dimensional problems (Mallat, 2016; Albawi et al, 2017). Convolutional neural networks use a convolution operation. The convolution operation is a mathematical operation for two functions that results in a third function that expresses the change in shape of one function by the other.

For different types of input data, there are different convolution variants, determined by the convolution kernel parameters. For two-dimensional convolution, on the example of image analysis, the kernel is defined by the image resolution and depth, the number of color channels, of the image. For one-dimensional convolution, the kernel parameters are the length of the input sequence, the number of elements in one line of the sequence, and the depth, the number of values per sequence step.

A deep neural network consists of layers of neurons with specified parameters. In one layer all neurons have the same convolution parameters. An example of one-dimensional convolutional neural network for sequence analysis using chemical formula is shown in fig. 4.2.1.

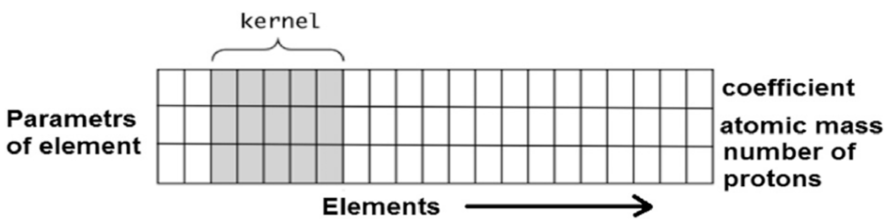


Figure 4.2.1. Example of one-dimensional convolutional neural network

The one-dimensional convolution layer is used to analyze two-dimensional sequences of data. This type of layer creates and uses a convolution kernel, which convolves with the layer input along one spatial or temporal dimension to produce a tensor of outputs. In the case of the formula analysis in Figure 4.2.1, the neuron, depending on the size of the convolution kernel, analyzes the parameters of each group of elements, moving from left to right.

Neural networks are flexible and scalable algorithms that can adapt to the data used in training. However, they are trained with a stochastic learning algorithm and adapt to the features of the training data during training. Therefore, even neural networks of the same architecture trained on the same data set, but started training with different parameters of weights, can find different variants of the optimal set of weights in each training, which in turn leads to different predictions.

To improve training results and reduce overtraining, an approach is used that is based on training on the same data and then combining multiple neural networks with different architectures. This is called ensemble learning and not only reduces prediction variance but can also lead to predictions that are better than any single model. An example of an ensemble of neural networks is shown in Figure 4.2.2.

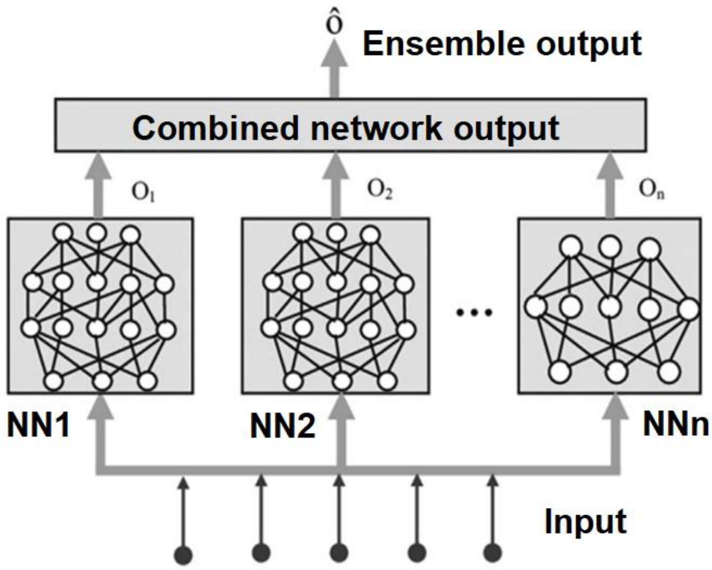


Figure 4.2.2. Example of an ensemble of neural networks (Alam et al, 2019)

Various methods are used in the training process to evaluate the learning quality and performance of the neural network. Prediction of the critical superconductivity temperature value for a chemical formula is a regression problem. To solve the regression problem, we use metrics: mean absolute error, mean square error, root mean square error and coefficient of determination.

The mean absolute error, mean absolute error, MAE, is a measure of the error between paired observations expressing the same phenomenon. A feature of MAE is its robustness to outliers in the data. Figure 4.2.3. shows the formula for calculation of mean absolute error MAE.

$$MAE = \frac{\sum_{i=1}^{n} |x_i - y_i|}{n}$$

where x - real values
y - predicted values
n- numbers of samples

Figure 4.2.3. Calculation of mean absolute error of MAE.

Mean squared error, mean squared error, MSE is calculated as the mean difference between the predicted and actual values. The result is always positive, regardless of the sign of the predicted and actual values, and the ideal value is 0.0. A quadratic value means that larger errors lead to larger errors than smaller ones, which means that the model is penalized for larger errors. Figure 4.2.4 shows the formula for calculating the mean square error MSE.

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (x_i - y_i)^2$$

where x - real values
y - predicted values
n- numbers of samples

Figure 4.2.4. Calculation of the mean square error MSE

The root mean squared error, RMSE, is a measure of the difference between the values predicted by the model or estimator and the observed values. It is calculated as the square root of the MSE, and large error values have a disproportionately large effect on the RMSE. Figure 4.2.5 shows the formula for calculating the RMSE.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (x_i - y_i)^2}{n}}$$

where x - real values
y - predicted values
n- numbers of samples

Figure 4.2.5. Calculation of RMSE standard deviation

The coefficient of determination, R2, is the proportion of variance in the dependent variable that is predictable from the independent variable or variables. It gives an estimate of how well the observed results are reproduced by the model based on the proportion of total variation in the results explained by the model. Figure 4.2.6 shows the formula for calculating the coefficient of determination R2

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(x_i - y_i)^2}{\sum_{i=1}^{n}\left(x_i - \underline{y}\right)^2}$$

where x - real values
y - predicted values
n- numbers of samples

Figure 4.2.6. Calculation of determination coefficient R2

4.2.4. Process of neural network model development and training

The processed data set was shuffled and randomly divided into 3 parts: 80% was the training sample, 10% was the test sample, and 10% was the validation sample. Each value within each sample was duplicated 5 times, and then the sample was shuffled. This operation was performed in order not to emphasize the peculiarities of the sample structure during training. Since the formulas had different numbers of elements in their composition, an element was added to the 10 elements, all parameters of which and the coefficient in the formula have the value 0.

After preprocessing, the original dataset containing 21263 formulas was divided into a training sample of 17010 unique formulas, a test sample of 2126 formulas, and a validation sample of 2127 formulas. Each formula in each subset was repeated 5 times, after which the sample was shuffled. As a result, the training sample contained 85050 formulas, the test sample contained 10630 formulas, and the validation sample contained 10635 formulas.

After dividing the dataset into samples, these data were used to train neural networks of different architectures. Since the formulas were a sequence of elements with specified parameters, the focus was on the neural network architectures based on LSTM and one-dimensional convolutional layers. The input of the neural network was fed with a processed formula consisting of 10 elements, each with 17 parameters: the coefficient of the element in the formula and 16 parameters of a chemical element from the periodic table of elements.

The models were trained in two stages. At both stages, the Adam optimizer was used. At the first stage, training was conducted using the loss function Mean Absolute Error, MAE with a learning rate value equal to 0.001. In the second stage, the previously trained model was trained using the Mean Squared Error, MSE loss function with a low learning rate equal to 0.0001.

4.2.4.1. Neural network models based on LSTM

Various architectures based on LSTM layers were considered during the study. The architectures and their names that gave the most accurate results are shown in Figure 4.1.7. Different activation functions and dropout values were investigated for these networks. The training results for these architectures are shown in Table 4.2.1.

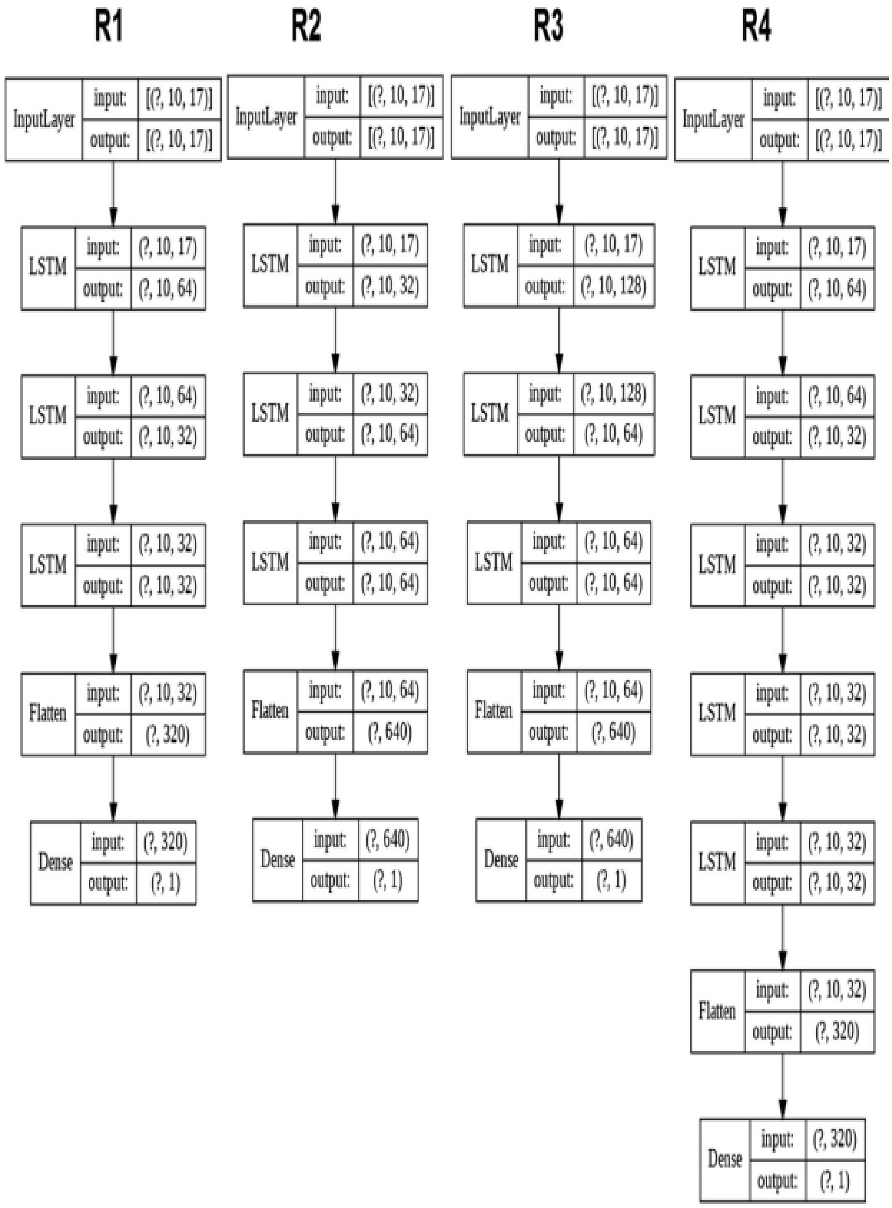


Figure 4.2.7. Model architectures based on LSTM layers

Table 4.2.1. Results of training models based on LSTM layers

Model name	***Dropout regularization***	***Activation function***	***MAE (degree ºK)***	***MSE (degree ºK2)***

R1	*0*	*none*	*4.8908*	*92.1871*
R1	*0*	*Relu*	*5.1268*	*100.8792*
R1	*0.05*	*none*	*4.8328*	*93.8598*
R1	*0.05*	*Relu*	*5.0758*	*97.2897*
R1	*0.1*	*none*	*4.9218*	*94.8925*
R1	*0.1*	*Relu*	*5.1089*	*96.2879*
R2	*0*	*none*	*4.9705*	*100.8791*
R2	*0*	*Relu*	*5.1798*	*110.8971*
R2	*0.05*	*none*	*4.9518*	*99.8791*
R2	*0.05*	*Relu*	*5.1209*	*107.9871*
R2	*0.1*	*none*	*5.0129*	*97.5872*
R2	*0.1*	*Relu*	*5.1408*	*100.1879*
R3	*0*	*none*	*4.9287*	*96.7898*
R3	*0*	*Relu*	*5.1287*	*110.8987*
R3	*0.05*	*none*	*4.9019*	*95.1982*
R3	*0.05*	*Relu*	*5.0791*	*105.2847*
R3	*0.1*	*none*	*4.9271*	*95.0879*
R3	*0.1*	*Relu*	*5.0971*	*101.2878*
R4	*0*	*none*	*4.9833*	*99.7972*
R4	*0*	*Relu*	*5.0917*	*100.1972*
R4	*0.05*	*none*	*4.9613*	*96.4879*
R4	*0.05*	*Relu*	*5.0796*	*104.1975*
R4	*0.1*	*none*	*4.9486*	*95.1479*
R4	*0.1*	*Relu*	*5.0579*	*109.1871*

According to the analysis of these results, the best results were obtained by neural network R1 of three LSTM layers, without neuron activation function on recurrent layers, with dropout value on each of the layers equal to 5%. The best results were obtained with the number of epochs equal to 300 and batch size equal to 200.

4.2.4.2. Univariate convolution based neural network models

The research also considered convolutional network architectures based on one-dimensional convolutional layers. For each layer, L2 regularization was applied with a value of 0.0001 to reduce the effect of overshooting. The architectures and their indices that gave the most accurate results are shown in Figure 4.2.8. For convolutional neural networks, training was performed at different kernel sizes. The convolution kernel size varied from 2 to 5. Training results for these architectures are shown in Table 4.2.2.

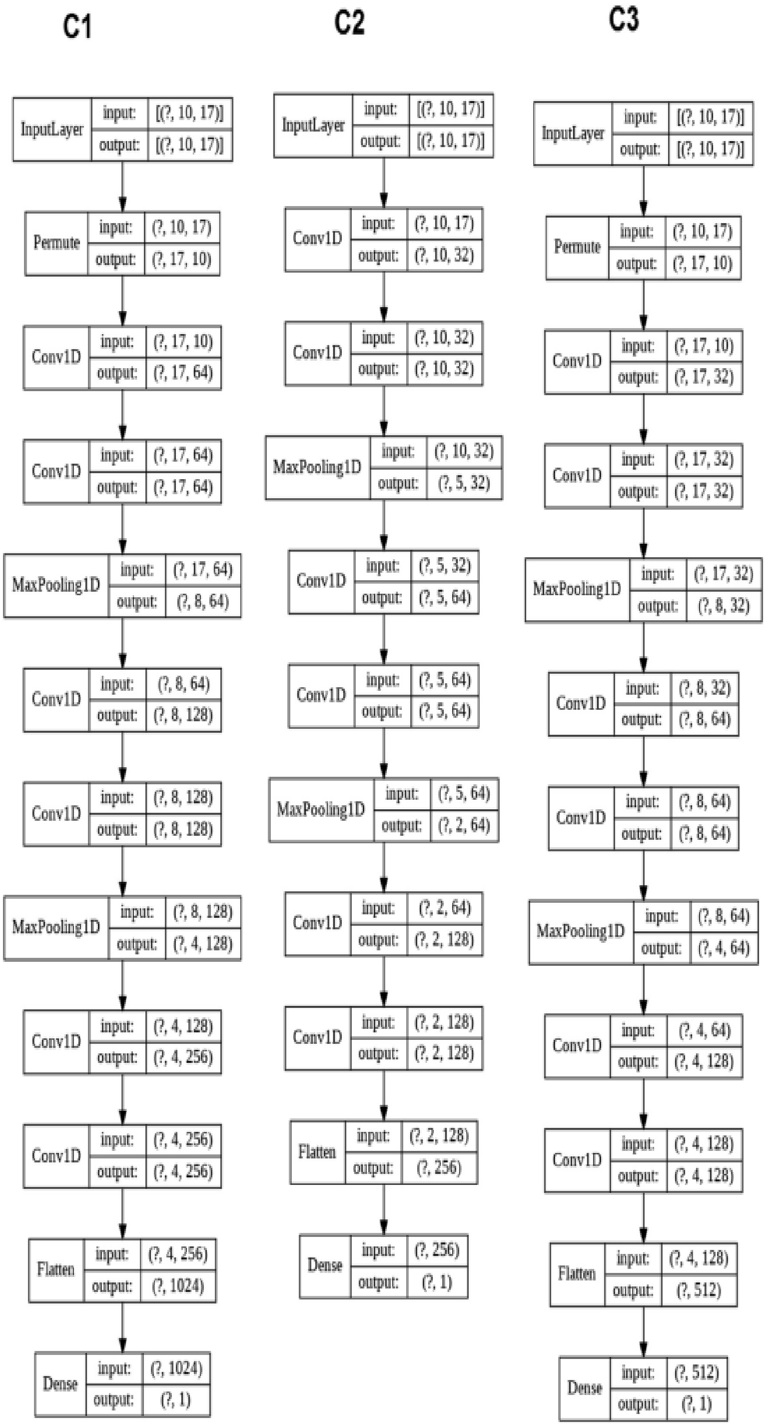


Figure 4.2.8. Model architectures based on one-dimensional convolution layers

Table 4.2.2. Results of training models based on one-dimensional convolutional layers

Model name	*Convolutional kernel size*	*MAE (degree ºK)*	*MSE (degree ºK2)*
C1	*2*	*4.7972*	*85.8867*
C1	*3*	*4.7287*	*85.4572*
C1	*4*	*4.7553*	*83.9879*
C1	*5*	*4.7843*	*89.4839*
C2	*2*	*4.8138*	*88.3587*
C2	*3*	*4.7628*	*89.7028*
C2	*4*	*4.8575*	*86.9126*
C2	*5*	*4.8052*	*85.8867*
C3	*2*	*4.8975*	*86.0501*
C3	*3*	*4.7559*	*85.7088*
C3	*4*	*4.8641*	*85.3812*
C3	*5*	*4.8384*	*88.4810*

The best results for convolutional neural networks were obtained with kernel size equal to three. The best results were obtained with the number of epochs equal to 300 and batch size equal to 200.

4.2.4.3. Cross validation approach to train neural network models

Cross validation training was used for variant architectures. The original dataset of 21263 values was divided 10 times into 90% of the training subset and 10% of the test subset. In addition, each value in each subset was duplicated five times, and then the subsets were mixed. The cross validated training subset contained 95685 formulas and the test subset contained 10630 formulas. The best training results using cross validation are shown in Table 4.2.3.

Table 4.2.3. Training results using cross validation layers

Model name	*MAE (degree ºK)*	*MSE (degree ºK2)*
R1	*4.8009*	*79.1871*
C1	*4.6608*	*78.0079*
C2	*4.7025*	*76.4987*
C3	*4.6989*	*79.7912*

All neural network architectures were trained on the best dataset with cross validation. All modules have two versions: adapted for MAE and adapted for MSE. The process of neural network training for MAE is shown in figure 4.2.9.

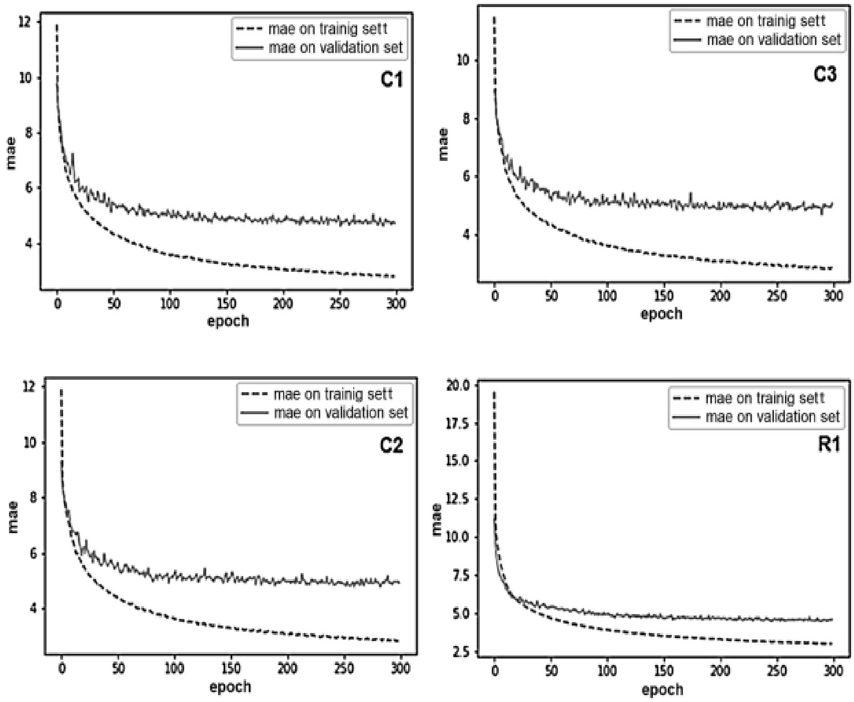


Figure 4.2.9. Neural network training process for MAE loss function for cross validation

The version of the cross validation dataset that gave the best training results and the neural networks trained on this data were saved and used to create the ensemble of neural networks.

4.2.4.4. Ensemble Neural Network Models

Since the C2 and C3 neural networks have a similar architecture, they were adapted to minimize MSE losses and combined into an ensemble. After training, the ensemble of C2 and C3 neural networks was inserted into the final ensemble. The C1 model was added in two versions: one adapted to minimize MAE losses and one adapted to minimize MSE losses. The final ensemble architecture of the pre-trained neural networks is shown in Figure 4.2.10.

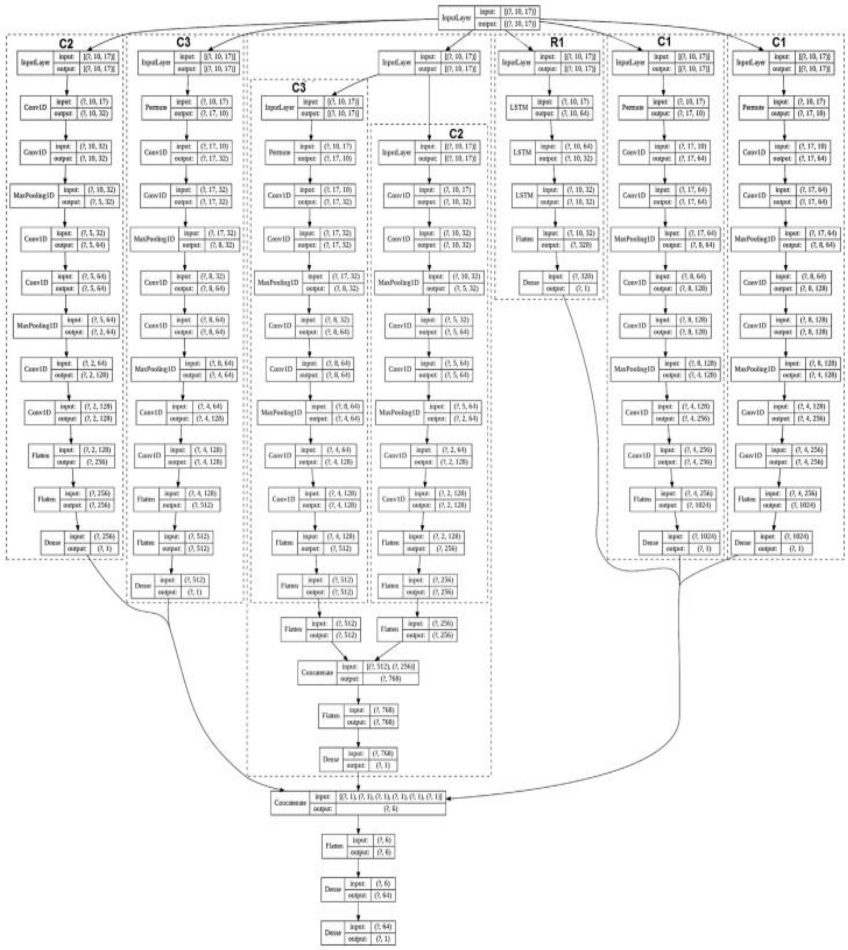


Figure 4.2.10. Ensemble architecture of pre-trained neural networks

In the ensemble, all outputs of all pre-trained models were combined and analyzed by a fully coupled layer. During the learning process, weights of the pre-trained models were frozen, only the last fully connected layer of 64 neurons was trained. The output values of 6 pre-trained networks were combined and sent to the full layer of 64 neurons with the "ReLU" activation function. The result of the full layer is used to estimate the critical temperature of the superconductor.

The final version of the ensemble was trained in 2 stages. In the first stage, training was performed using the mean absolute error loss function, MAE, with a training rate value equal to 0.00001. In the second stage, the previously trained model was trained using the Mean Squared Error loss function, MSE with a low learning rate equal to 0.0000001. The low learning rate can be explained by the fact that the ensemble consists of already pre-trained models. The best results were obtained with the number of epochs equal to 20 and batch size equal to 200. The ensemble training process to minimize MAE losses is shown in Figure 4.2.11.

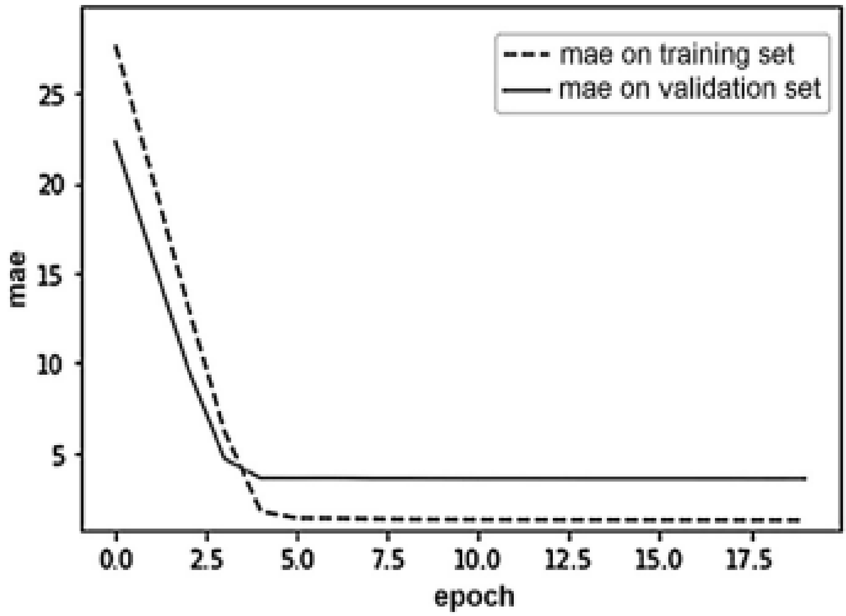


Figure 4.2.11. Ensemble training process to minimize MAE losses

The error of this ensemble after training is 4.068 for MAE loss, 67.272 for MSE loss. After minimizing, MSE, loss, the coefficient of determination, R2, was calculated. The coefficient of determination, R2, is 0.923. Also, after minimizing the MSE loss, the root-mean-square error, RMSE, was calculated. The RMSE is 8.202. The final ensemble of neural networks has 1330247 trainable parameters.

4.2.5. Results

This study examined the application of neural networks to the analysis of superconductor tempers. The analysis was based on the properties of chemical elements and their coefficients in the chemical formula. As a result, neural networks with convolutional and recurrent architecture were trained. An ensemble of neural networks was created as a combination of the best choices of pre-trained neural network architectures.

The accuracy scores of the ensemble after training are shown in Table 4.2.4.

Table 4.2.4. Post-training accuracy of the ensemble compared to its constituent models.

Model name	*RMSE (degree ºK)*	*R2*	*MAE (degree ºK)*
Ensemble R1, C1, C2, C3 networks	*8.202*	*0.923*	*4.068*
R1	*8.899*	*0.892*	*4.801*
C1	*8.832*	*0.897*	*4.661*
C2	*8.749*	*0.906*	*4.703*
C3	*8.932*	*0.890*	*4.699*

Table 4.2.5. Prediction results for this ensemble compared to previously developed algorithms.

Algorithm	*Author*	*RMSE (degree ºK)*	*R2*	*MAE (degree ºK)*
Multiple regression (Hamidieh, 2018)	*Kam Hamidieh*	*17.6*	*0.74*	-
XGBoost (Hamidieh, 2018)	*Kam Hamidieh*	*9.5*	*0.92*	-
XGBoost (Hamidieh, 2018)	*Abdulkadir Karacı*	*9.091*	*0.928*	-
Hybrid neural network (Shaobo et al, 2019)	*Shaobo Li*	*9,141*	*0.899*	*5.023*
Proposed model	*Authors of this dissertation*	*8.202*	*0.923*	*4.068*

The RMSE of the developed ensemble is less than the minimum RMSE error value of the previous algorithms. The RMSE value of the ensemble from decreased by 0.889.

The coefficient of determination is larger than that of the previous algorithms. The mean absolute error is less than the MAE of the previous algorithms. The R2 of the ensemble from this study decreased by 0.005.

The mean absolute error of the developed ensemble is less than the minimum error MAE of the previous algorithms. The mean absolute error of the ensemble from this study decreased by 0.937.

Although the ensemble of neural networks lost exactly the coefficient of determination, the MAE and RMSE values decreased significantly. Comparison of the change in MAE and RMSE values with the value of the coefficient of determination allows us to consider the decrease in the value per coefficient of determination as insignificant compared to the improvement in accuracy and decrease in MAE and RMSE values.

The developed neural network ensemble algorithm, like the previous algorithms presented for comparison, is based on the same dataset. This data set includes only the chemical formula of the superconductor and its critical temperature. However, many substances can change their internal structure depending on many factors. Adding information about the atomic structure of the material under study to the model could greatly improve the quality of models designed to analyze chemical compounds.

4.2.6. Conclusion

In this subsection "Neural network-based prediction of the critical superconductivity temperature for superconducting materials based on their chemical formula", an ensemble of neural networks was developed to predict the critical temperature of superconductors. The input data for this neural network model is only the chemical formula of the material. Sorting the chemical elements in the formula by the element number in the periodic table of chemical elements allowed the neural network to concentrate more actively on the actual parameters of the chemical elements, rather than on the features of their representation in the chemical formula.

Since the chemical formula is a sequence of parameters, recurrent and convolutional neural network algorithms were used in this neural network model. The combined use of these algorithms allowed a high accuracy of the target parameter, the critical temperature for the chemical formula. The proposed method based on the neural network model can be useful for the search of high-temperature superconductors.

4.3 Neural network-based prediction of the reduced glass transition temperature of metal alloys based on a neural network.

The reduced glass transition temperature Trg is an important parameter for glass formation. Trg describes the glass formation in materials and the behavior of materials during the transition between solid and liquid states and is an important parameter for materials analysis, development and production process. This subsection describes the process and results of research on the development of a system for predicting the glass transition temperature of metal alloys Trg based on recurrent neural network algorithms. The developed system can predict the reduced glass transition temperature Trg of metal alloys with high accuracy based on the analysis of their chemical formula. The accuracy was evaluated by 3 metrics: MSE, RMSE, MAE. The values obtained are as follows: MSE - 0.000678, RMSE - 0.0260, MAE - 0.01835.

4.3.1. Introduction

Glass forming ability, GFA, describes the formation of glass in materials. GFA can be represented by many parameters (Lu et al, 2000). Reduced glass transition temperature Trg is one of the commonly used GFA parameters and describes the behavior of materials during the transition between solid and liquid states and is an important parameter for material analysis, development and manufacturing process. Control of this parameter can be used to control the crystal lattice structure of materials and control physical and chemical parameters of materials. The knowledge of this parameter can allow you to control the properties of materials made by casting methods. Materials with different properties can be used in different spheres of human activity: social, scientific, medical, etc.

The reduced glass transition temperature Trg is based on the glass transition temperature Tg and the liquidus temperature Tl. The calculation of the reduced glass transition temperature Trg is shown in Figure 4.3.1.

$$Trg = Tg \,/\, Tl$$

Figure 4.3.1. Calculation of the reduced glass transition temperature

The reduced glass transition temperature Trg depends on the composition of the material. Thus, the analysis of the chemical composition of the material can be used to calculate a given Trg.

This study is based on a public dataset on the reduced glass transition temperature Trg for metal alloys (Wu et al, 2016) and investigates the possibility of predicting the reduced glass transition temperature Trg of an alloy based on the chemical formula of the alloy using neural networks. Creating a model to predict Trg for an uncertain alloy will allow these values to be used to estimate the GFA of the material directly or as input to another model to then predict the GFA of the material with higher accuracy.

4.3.2. Materials

4.3.2.1. Data Analysis and Processing

This study is based on an open data set of reduced glass transition temperature Trg for metal alloys. The dataset contains 585 samples with 23 columns of parameters.

The dataset contains two columns of material composition information. The first column is the general composition and description of the chemical formula of the alloy. This is the input parameter for the neural network model.

The second column is the element of the chemical formula with the highest order number. This column simply duplicates information that is not used in the neural network model.

The reduced glass transition temperature Trg is in the third column and is the target output parameter of the neural network model. It is used as a rough predictor for the GFA.

Columns four through 23 are MAGPIE characteristics that were calculated from the material composition column. These parameters give values such as properties averaged over the material composition, as well as characteristics that apply only to most of the elements in each alloy (Ward et al, 2015). These parameters are used in the training process of the neural network but are not used in the final model.

The output parameter Trg values are in the range of 0.223 and 0.688. The largest number of alloys have a Trg value close to 0.6.

This dataset is divided into a training dataset, a validation dataset, and a test dataset. The training dataset is 80% of the full dataset, 468 samples. The test dataset is 10% of the full dataset, 58 samples. The validation dataset is 10% of the full dataset, 59 samples.

The original data set has a small number of samples and random division may have statistical anomalies in the distribution of the data. Using the Kolmogorov-Smirnov test [Wayne et al, 1990] helps avoid problems with the distribution of the data. The data were randomly partitioned and compared to the distribution of the original data to obtain the minimum value of the Kolmogorov-Smirnov test. The distribution with the minimum value of the Kolmogorov criterion and the most similar to the original distribution is shown in Figure 4.3.2.

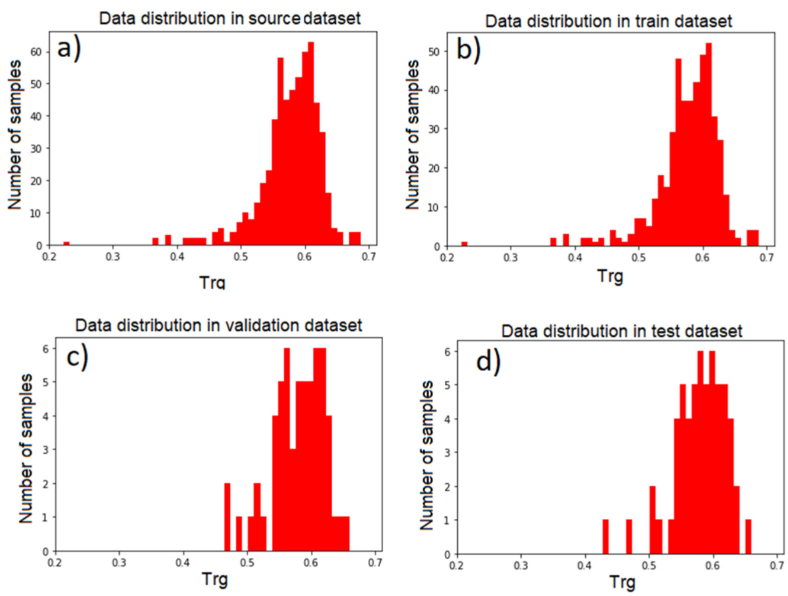


Figure 4.3.2. Distribution of the reduced glass transition temperature in the initial, training, test and validation data sets

Figure 4.3.2. denotes distributions:

(a) The distribution of the data in the original data set;

(b) Distribution of data in the training dataset;

(c) The distribution of the data in the test data set;

(d) The distribution of data in the test data set.

Formulas in the dataset are represented by lines containing formula descriptions in classical chemical format, e.g., Cu47Ti33Zr11Ni2Sn6Si. During data processing, each formula is broken down into two lists. The first list contains the elements of the formula. The second list contains the chemical coefficients of the formula.

Each unique element from list 1 is analyzed by the tokenizer (Ramasubramanian et al, 18) and receives a unique integer label.

The chemical coefficients in a formula can have significantly different values. The neural network will not give good results for raw values of chemical coefficients. To avoid scattered values, these parameters are standardized (Shanker et al, 1996).

A chemical formula can have up to 9 elements in its composition. The neural network cannot analyze data with dynamic length. Each formula with a small number of elements is complemented by up to 9 zeros at the beginning of the formula in both lists.

4.3.2.2. Neural network design and training

A neural network based on LSTM recurrent layers was developed for data analysis (Understanding LSTM Networks)

The neural network has 2 input layers, which receive data about the chemical composition of the alloy.

The first input layer receives data on the chemical elements of the alloy, designated by integer labels. They are analyzed by the embedding layer (Vo et al, 2019), and each chemical element label is described by a tensor of 64 elements. At the output of the embedding layer, a two-dimensional array of 9x64 tensors is formed.

The second input layer receives standardized coefficient values of chemical elements of the formula. It is merged with the two-dimensional 9x64 tensor array by the concatenate layer. The resulting 9x65 array is analyzed by two consecutive LSTM recurrence layers. The output of the recurrence layers is a 9x128 tensor array.

This 9x128 array is converted into a tensor of size 1152 and subjected to a dropout regularization (Yarin et al, 2017) to eliminate the overfitting effect (Hawkins et al, 04). The resulting tensor is passed to the last layer to predict the target features. The last output layer does not use any activation function.

The neural network uses the mean square error MSE as a loss function and the mean absolute error MAE as an additional metric. The training uses an Adam optimizer, which has a progressively decreasing learning rate. The learning rate starts at 0.01 and decreases to 0.0001.

These parameters were optimized and obtained by TensorFlow HParams Dashboard grid search.

The training is done in two stages. In the 1st stage, the neural network is trained to predict every available 21 features of the chemical formula: the reduced glass transition temperature Trg and 20 MAGPIE features. In the second stage the neural network is trained to predict only the reduced glass transition temperature Trg.

This training allows the neural network on the 1st step of the training to get the most complete understanding of the structure of the analyzed data, which allows to get the best results of the target metrics on the 2nd step.

Neural network architecture on the 1st step of training is presented in figure 4.3.3.

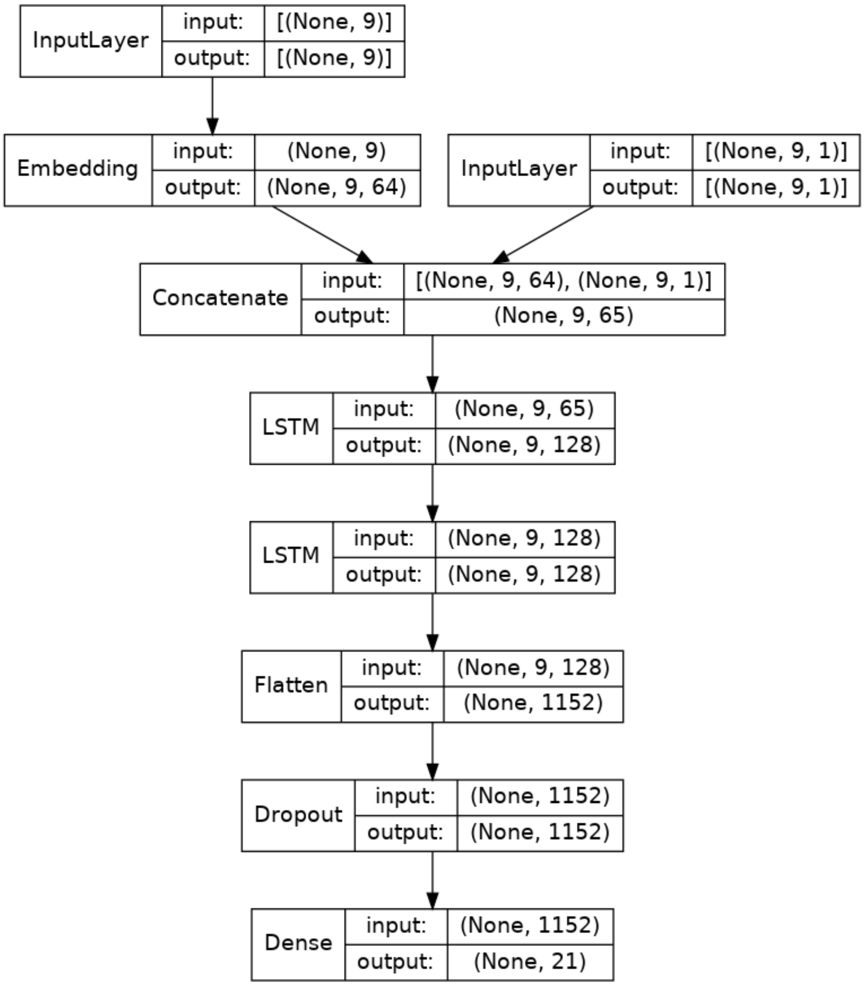


Figure 4.3.3. Neural network architecture on the 1st stage of training

Training on the 1st stage is performed for 1000 epochs. The process of changing the value of loss function MSE is shown in figure 4.3.4.

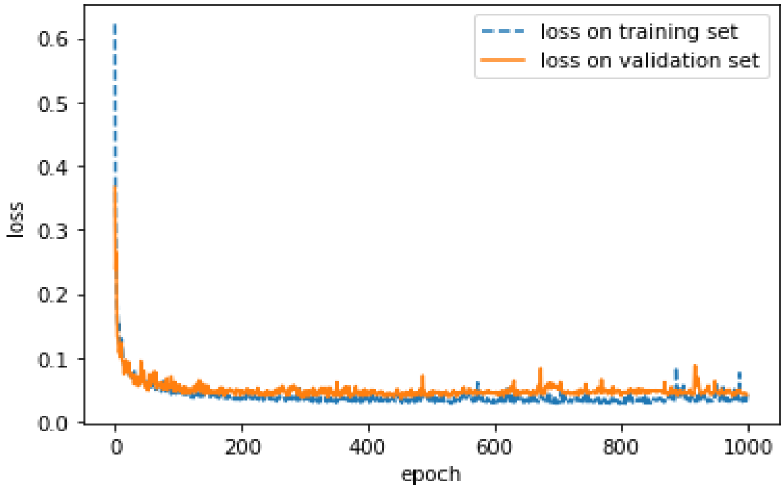


Figure 4.3.4. Change of MSE loss function value at the 1st stage of training

The best obtained values of MSE loss function and MAE metric for 21 output features at the 1st stage of training are as follows: MAE - 0.0857, MSE - 0.0348. This is the sum of the losses for each of the 21 output features. This model has 264.725 total parameters. This version is retained.

After step 1 the best version of the neural network is reconstructed. The output layer of the reconstructed model is removed. In the second step a new output layer with one output value is added to the model. The training parameter for the pre-trained model is disabled for the first step of step 2, because the new untrained layer may disturb the weights of the pre-trained neural network. In the first step of step 2, only the last layer of the model is trained. The model is trained for 100 epochs at a learning rate of 0.01. At this stage the model has 241.665 total parameters and 1.153 trainable parameters.

The architecture of the neural network in the second stage of training is shown in Figure 4.3.5.

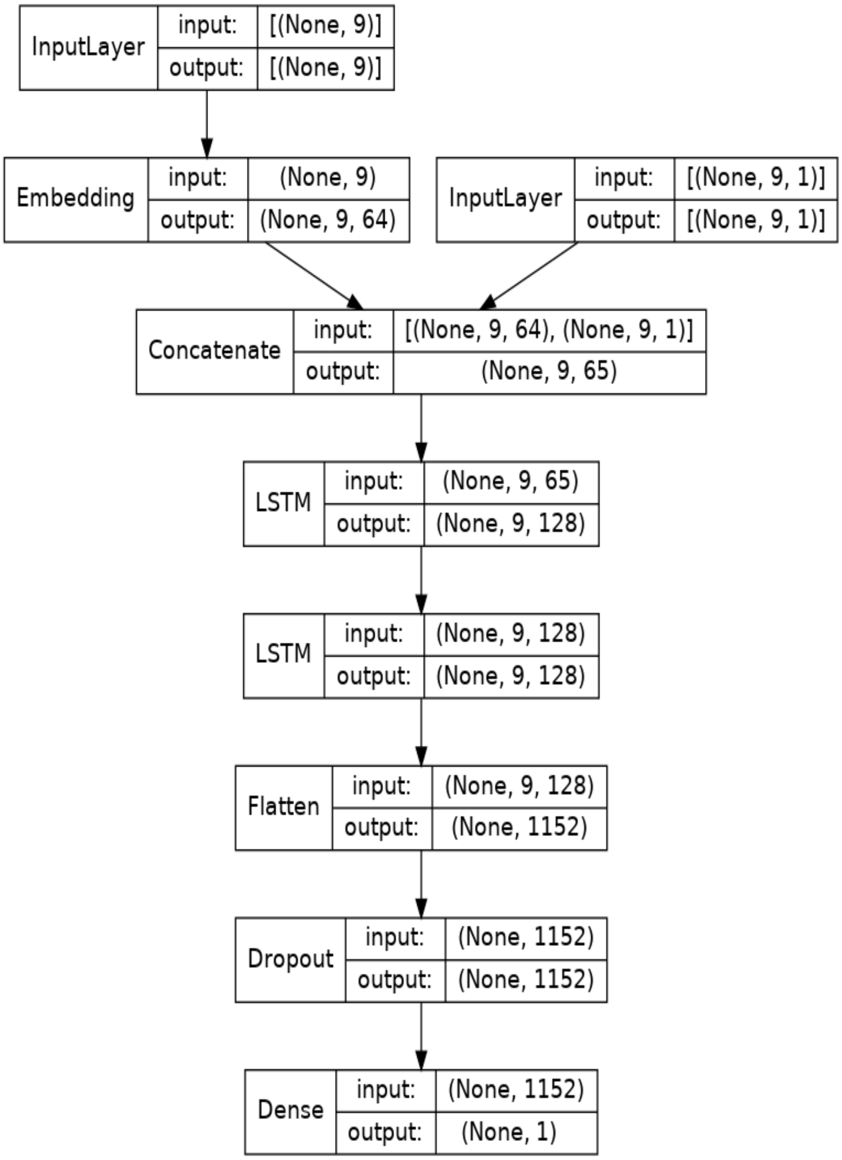


Figure 4.3.5. Neural network architecture in the 2nd stage of training

The first step of training at the 2nd stage is done in 100 epochs. The process of changing the value of loss function MSE is shown in figure 4.3.6.

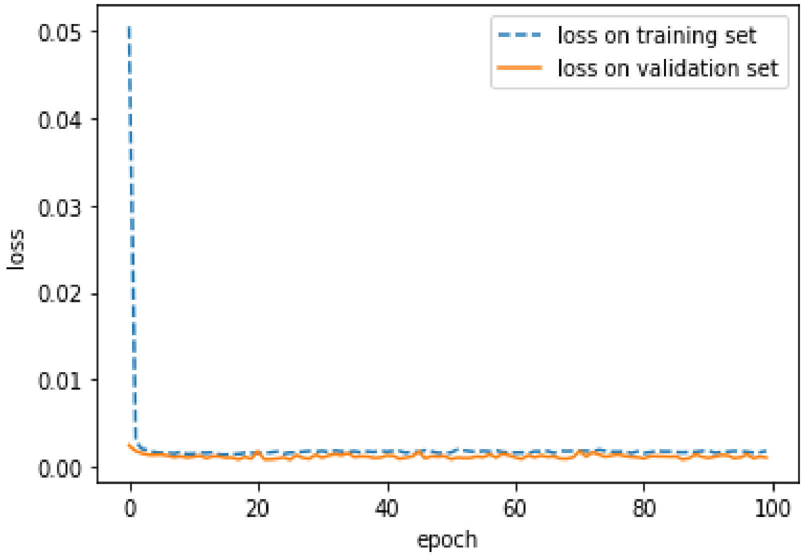


Figure 4.3.6. Change of MSE loss function value at the 1st step of the 2nd stage of training

The best values of MSE loss function and MAE metric for 1 output feature at the 1st step of the 2nd stage of training are as follows: MAE - 0.02098, MSE - 0.000799. This version is retained.

After the 1st step of the 2nd stage the best version of the neural network is restored. The training parameter for the full model is turned on. The model is trained for 100 epochs with a low learning rate of 0.0001. At this stage the model has 241.665 trainable parameters.

The best values of the MSE loss function and MAE metric for 1 output feature at step 2 of the 2nd stage of training are as follows: MAE - 0.01835, MSE - 0.000678. This version is retained. This is the final version of the model.

4.3.3. Results

This dataset is small, and dividing it into test, validation, and training samples randomly creates anomalies in the distribution of the data. Using a dataset with a larger data size can significantly improve the performance of the neural network.

In addition, the quality of neural network training can be improved by using additional data.

Another way to obtain the most robust model can be to further balance the data by clustering it. This approach, however, will reduce the quality of the model's target metrics, but will allow it to give the same quality on the target metrics on any distribution of data.

4.3.4. Discussion

This subsection “Neural network-based prediction of the reduced glass transition temperature of metal alloys based on a neural network” developed a recurrence layer based neural network model to analyze the chemical formulas of alloys and predict their reduced glass transition temperature Trg. Training was performed in 2 stages. In the 1st stage of training, the model received the fullest available information about the chemical compound. At the 2nd stage, the model trained at the 1st stage was improved to predict only the target parameter - reduced glass transition temperature Trg.

The model works with the processed data. Names of chemical elements were subjected to tokenization. The coefficients of chemical elements were subjected to standardization. Using tokenization without additional allowed to show that neural networks can be used for prediction even if only information about the structure of the material is known, but the properties of its components are unknown. This approach can be used to analyze materials whose structure is incompletely described at different scales.

The combination of data processing algorithms and neural network learning algorithms resulted in the best metric values. The best obtained values of MSE loss function and MAE metric are as follows: MAE - 0.01835, MSE - 0.000678. The calculated value of RMSE is 0,0260.

4.3.5. Conclusion

This subsection “Neural network-based prediction of the reduced glass transition temperature of metal alloys based on a neural network” shows the process of developing a recurrent layer based neural network to analyze the chemical formulas of alloys in order to predict their reduced glass transition temperature Trg. As a result, the final neural network model has metric values: MAE - 0.01835, MSE - 0.000678, RMSE - 0.0260.

The possibilities and peculiarities of using neural networks for the analysis of chemical formulas of dynamic size with a wide variation of their coefficients are demonstrated. Since the analyzed data set has a small size, the work provides examples of data anomalies elimination and processing.

4.4 Neural network-based prediction of the composition of paper-like materials based on the specified chemical and physical parameters

This subsection describes the development of a neural network to analyze the composition of paper-like materials based on specified chemical and physical parameters. The development of new paper-like materials is a long and expensive process. The use of material parameter modeling can make it possible to speed up and cheapen the production process of new materials. For this purpose, a neural network model is proposed in this section. It is required to predict 5 parameters. Five models are trained. Each model has 15 input parameters. Each model has its own predicted parameter describing the physical and chemical composition of the material approximating the properties of the given input physical and chemical input parameters. On the available data for calculation of concentrations of different particle sizes: nano solid fibers NSF-0,1; micro solid fibers MSF-0,25; macro solid fibers MSF-0,4; ultra-solid fibers USF-0,6 the values of average absolute error of calculation of concentrations equal to 4.1%; 5.3%; 23.4%; 14% respectively were obtained. For the calculation of Al2(SO4)3 concentrations, the value of the mean absolute error is 1.5%.

4.4.1. Introduction

Paper-like materials consist of different types of particles and are produced over a long period of time. The process of developing such new materials is a lengthy experimental study due to the length of time it takes to produce different formulations and test them, but the results of such a study will allow the production of cheap materials with different properties from cellulose fibers (Zhu et al, 2016; Mosier, 2005). Such materials can be used in various applications and exhibit different physical and chemical properties due to different concentrations of their constituent cellulose fibers of different shapes and sizes (Abdul et al, 2014).

These materials are eco-friendly and can be used in various social, scientific, technical fields of aspects of the human activity. Paper-like materials can have different physical and chemical properties, depending on the shape and size of the cellulose fibers used in their creation, as presented in this dataset. Also, such materials are developed based on environmentally friendly materials and are safe for human use for a long period of time in scientific, social, medical, and technological fields of activity. However, finding a material with the desired properties and predicting the material composition based on the desired properties is a challenging task (Lavoine et al, 2012; Klemm et al, 2005). For this process, this section proposes to train and use a neural network model.

4.4.2. Materials

The data for the analysis is a table with experimentally obtained results from experiments with paper-like materials with known compositions. After purging the uncorrected results, the table contains data on the properties and structure of paper-like materials for 45 tested samples. Each sample is described by 20 parameters.

The table is divided into a training subsample of 36 samples, a test subsample of 4 samples, and a validation subsample of 5 samples.

The parameters NSF-0.1; MSF-0.25; MSF-0.4; USF-0.6; Al2(SO4)3 determine the structure of the material and will be used in the training process as output predictive parameters.

The coefficients in parameters NSF-0,1; MSF-0,25; MSF-0,4; USF-0,6 define the size of used fibers. The values of parameters NSF-0,1; MSF-0,25; MSF-0,4; USF-0,6 define a concentration of these fibers in a final material. In total, these parameters give 100%. The parameters were normalized to a range of 0 to 1.

The parameter Al2(SO4)3, describes the concentration of aluminum sulfate used in production. The value is in the range of 0 to 15%. The parameter is normalized to a range of 0 to 1.

Parameters airflow resistance, Pa; airflow resistance, mm water column. ; particle slip coefficient, max; particle slip coefficient, min; cleaning efficiency, max; cleaning efficiency, min; porosity, ml/min; roughness, ml/min; porosity by Herlel, sec; modulus of elasticity, MPa, transverse; modulus of elasticity, MPa, longitudinal; tensile strength, MPa, transverse; tensile strength, MPa, longitudinal; strain, %, transverse; strain, %, longitudinal; tensile length, m, transverse; tensile length, m, longitudinal;

forming index determine material properties and will be used in the model as input parameters. These parameters were also standardized during data preparation and analysis.

The table property columns porosity, ml/min; roughness, ml/min; and Herlei porosity, sec contain uncorrected measured values that cannot be used to train the machine learning system. It is impossible to exclude rows with incorrect measured values due to the extremely small amount of data for analysis, 45 examples. Excluding individual rows would have a much more negative impact on the final model than eliminating these columns from the calculations. These columns are excluded from the calculations. If the amount of experimental data increases, these columns can be included back into the calculation, and the rows with incorrect values are excluded from the calculation.

4.4.3. Methods

Figure 4.4.1 shows the neural network architecture for predicting the selected parameter. At the end of training 5 such neural network models will be trained, 1 for each target parameter of composition.

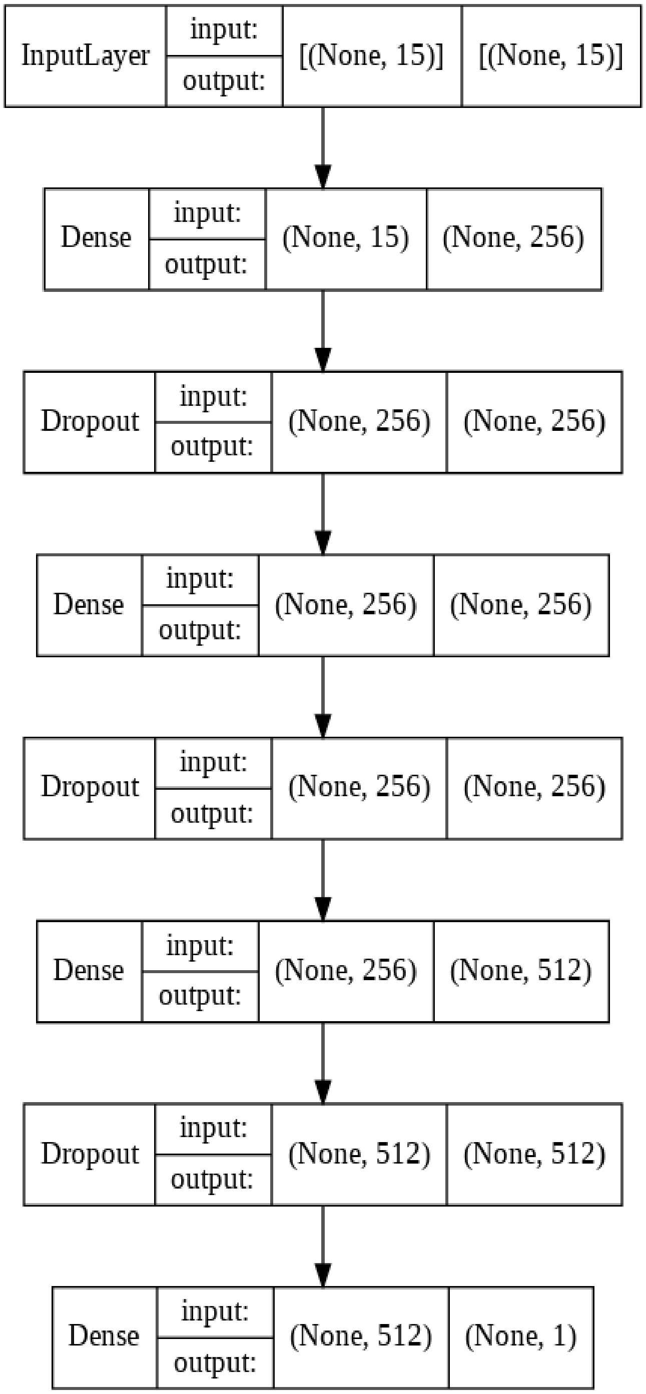


Figure 4.4.1. Neural network architecture

During model training, because of the small amount of data, the starting weights of model neurons have a significant impact on the final result. Model starting weights are generated based on a normal distribution. To use this fact to improve the target performance of the model, the model with the same structure is trained 100 epochs 30 times for each parameter. The model with the best performance is retained. The mean absolute error is used as the loss function. The optimizer is Adam with a learning rate of 0.001.

This approach can be used to predict properties based on composition and vice versa, composition based on required properties. However, due to significant data limitations, predicting properties based on composition with satisfactory accuracy is not possible because the possible variations of 5 composition elements are not sufficient to predict 15 parameters of a given material. This problem can be eliminated by expanding the data set used to train the models.

More promising is the variant of predicting material structure on the basis of required parameters, because 15 parameters give a much larger number of different variations of values than 5 parameters. This fact makes it possible to predict the 5 parameters of the material structure, despite the significant limitation in the data.

The influence and correlation of the corresponding elements of the composition on the final properties are investigated, taking into account the factor that the sum of values of columns NSF-0,1; MSF-0,25; MSF-0,4; USF-0,6 must be 100%. Each parameter is calculated by the model independently Since any machine learning model has an error in the predicted readings, the column with the least influence on the target parameters is filled with the value supplementing the composition to 100%. That is, 3 parameters out of 4 are predicted. The last parameter, which has the worst correlation value and is the most difficult to predict, is calculated by augmenting to 100%. The neural network is not used to calculate the concentration of this parameter.

The parameter Al2(SO4)3 is calculated by a different model separately from NSF-0,1; MSF-0,25; MSF-0,4; USF-0,6.

4.4.4. Results

The models for each predicted parameter were trained independently.

Due to the small amount of data, the initial weights of the neural networks have a significant impact on the results. The model with the same structure is trained 100 epochs 30 times for each parameter to improve the target performance. The model with the best performance is retained. The experiment is repeated for standardized and normalized parameters.

Due to the small amount of data at the first epochs, incorrect readings of the accuracy of the neural network are possible. The analysis of the quality of model performance begins at epoch 10 of training. Epochs before 10 are not used in the analysis of the neural network performance. After epoch 10 the neural network reaches the learning plateau and the quality of the model on the test data does not change jumps. Figure 4.4.2 shows the learning process of neural networks and the change of mean absolute error after 10 epochs for each of the target parameters to predict normalized data for training and test data.

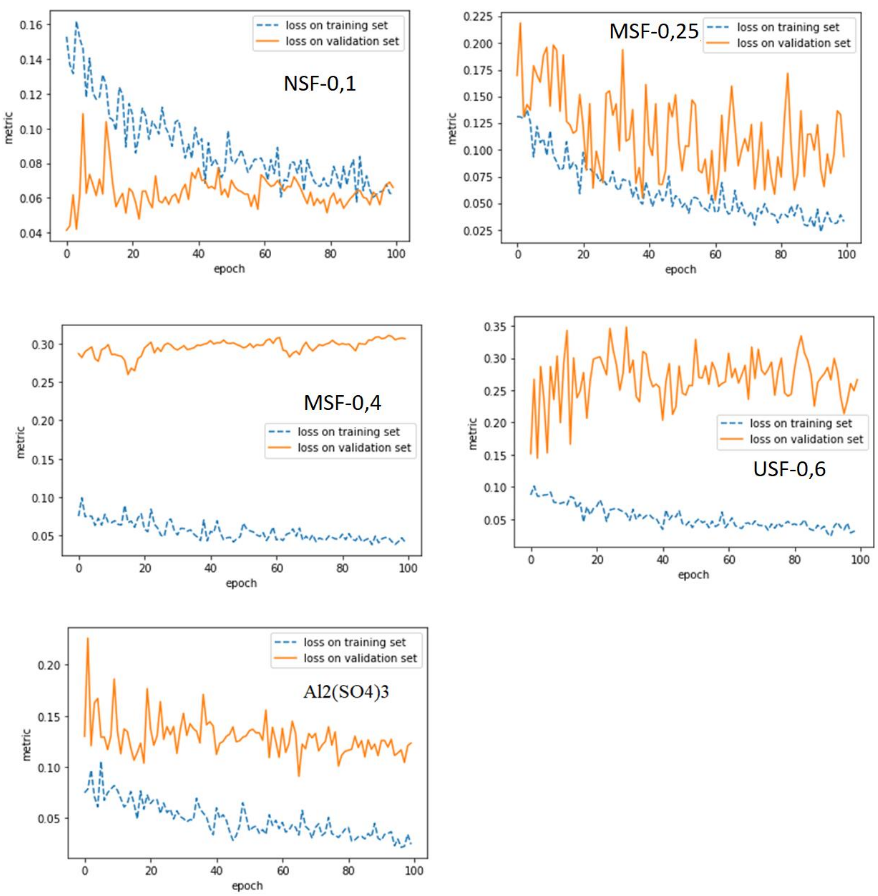


Figure 4.4.2. Process of training of neural networks and change of mean

absolute error for selected parameters on normalized data

In the process of training for normalized parameters NSF-0,1; MSF-0,25; MSF-0,4; USF-0,6 we managed to achieve values of mean absolute error 0.041; 0.053; 0.26; 0.14 respectively, which correspond to prediction accuracy in absolute values of 4.1%; 5.3%; 26%; 14%.

The parameter MSF-0,4 was predicted with the highest error. This means that in the available data the dependence between input parameters and a given output parameter is minimal. However, it is present because if there were no dependence at all, the error would be 0.5. It is likely that more data will be needed to identify this dependence more accurately.

As a result, the model for the calculation of MSF-0,4 is excluded from the final calculation and this parameter is calculated by adding it to 100% on the basis of the calculated parameters NSF-0,1; MSF-0,25; USF-0,6. With this approach, the error for the calculation of MSF-0,4 will be between 14% and 23.4% in the worst case forecast, when all parameters NSF-0,1; MSF-0,25; USF-0,6are calculated with maximum error. Even in the worst case of this approach, the forecast quality improves by 2.6%.

In the learning process for the standardized parameters $Al2(SO4)3$ it was possible to achieve a mean absolute error value of 0.09, corresponding to a prediction accuracy of 1.35% in absolute values, since the initial range for this parameter before normalization was 0-15%.

4.4.5. Conclusions

In this subsection "Neural network-based prediction of the composition of paper-like materials based on the specified chemical and physical parameters", the capabilities of neural networks to predict the composition of paper-like materials based on the required chemical and physical parameters are discussed. Prediction accuracy values of 4.1%; 5.3%; 23.4%; 14% were achieved for NSF-0.1; MSF-0.25; MSF-0.4; USF-0.6, respectively.

In the available data, the relationship between the physical and chemical parameters and the MTB-0.4 output parameter is poorly traced. The correlation between the physical and chemical parameters and the output parameter of USF-0.6 is somewhat better traced. However, correlations between input and output parameters of MSF-0,4 and USF-0,6 are observed, hence with the increase of the analyzed data volume the probability of detection of dependencies and correct model training is high. Also, the use of additional calculations may allow to reduce the error of calculations without increasing the dataset.

The problem of dataset size has been solved by multiple training of the neural network. Randomly generated weights were used to find different features of the data. The model with the best loss function score for each feature was retained. The model with the worst loss function score for parameter MSF-0.4 was excluded from the calculations. The calculations were performed by adding the concentration to 100%. This approach allowed an accuracy gain of 2.6% for this parameter.

4.5 Neural network for generation of composition and parameters of metal alloys based on specified range of known and unknown parameters

This subsection describes the development of a generative-adversarial neural network for generating metal alloy compounds with given parameters. The resulting alloy is described by 19 parameters. 5 parameters describe the properties of the alloy. 14 parameters describe the composition of the alloy. The parameters are normalized to a range of 0 to 1 before being processed by the neural network. The generator in the generative-adversarial network has 4 input layers. The first input layer receives noise to generate different realistic parameters at the same input values. The second input layer is a mask with descriptions of known and unknown parameters. The number of unknown parameters can be up to 10. The third input layer receives the minimum allowable values of the parameters specified by the user. The fourth layer of the generator receives the maximum allowable values of the parameters. The generator output generates 19 parameters describing the alloy based on the parameters specified at the input. The discriminator checks the validity of the prediction made by the generator. The discriminator has 4 inputs. The first input layer of the discriminator receives the forecast made by the generator. The other 3 inputs come from the generator input layers 2,3,4. The generator-adversarial neural network is able to generate the composition and properties of alloys with a mean absolute error value of 0.082 units relative to the normalized range of test data parameters, i.e. with an accuracy of 91.8% relative to the real value

4.5.1. Introduction

Material design is a complex but important task. Sustained scientific, medical technical and social progress requires new materials for use in various human activities (Himanen et al, 2019; Dresselhaus et al, 2005; Stach et al, 2021; Pintsuk et al, 2019). One of the important areas of materials development is the development of new metallic alloys (Himanen et al, 2019; Pintsuk et al, 2019; Kipouros et al, 2006). A challenging aspect is the development of a mathematical model for several metallic alloys that takes into account the interaction of the various components of the alloy and the influence of these interactions on the final material properties (Kipouros et al, 06; Hardian et al, 2020).

The development of artificial intelligence and machine learning algorithms has made it possible to automate the search for dependencies between alloy components and to automate the calculation of the influence of components on the final material properties (Himanen et al, 2019; Ball, 2019; Sha et al, 2020; Wei et al, 2019; Huang et al, 2020). However, the use of machine learning and artificial intelligence has long been limited to predicting the properties of alloys based on their composition, and the automatic creation of materials based on their chemical composition and required parameters has long remained a challenging task (Himanen et al, 2019; Huang et al, 2020).

The use of generative-adversarial neural networks can allow the generation of data according to given parameters and constraints (Creswell et al, 2018; Wang et al, 2017). This approach can also be applied to the design of materials with given parameters and properties.

This subsection describes the possibilities of using generative-adversarial networks to develop metal alloys with user-defined properties and composition constraints. The subsection describes the neural network architecture, the neural network training algorithm, the data preprocessing algorithm and the results obtained.

4.5.2. Materials

This paper uses an open-access dataset describing the properties and composition of metal alloys (Mechanical properties of low alloy steels). The dataset contains 915 lines with descriptions of metal alloys with different compositions under different conditions. Each alloy is described by 20 parameters: alloy code, C, Si, Mn, P, S, Ni, Cr, Mo, Cu,V, Al, N, Ceq, Nb + Ta, Temperature (°C), 0.2% test stress (MPa), tensile strength (MPa), elongation (%), area reduction (%).

Alloy code - description of the alloy grade. This parameter was important in dividing the data set into a training data set, a test data set, and a validation data set. The alloy grades that were in the training dataset were not added to the test and validation datasets. There are a total of 95 alloy grades in the dataset. The training dataset contains descriptions of 76 alloys and 729 samples, the test dataset contains 9 alloys and 88 samples, and the validation dataset contains 10 alloys and 98 samples. After splitting into training, test and validation datasets, the "Alloy Code" column is removed from the dataset because it does not contain information that can be used in the generative-adversarial neural network. Only the numerical parameters of the alloys remain.

The range of values for the remaining 19 parameters varies considerably, which would complicate the analysis of such data and the training of the neural network. The range of values for each parameter is normalized to a range of 0 to 1 based on the training dataset. Test and validation datasets were not used in calculating the parameters for normalization. This was done to avoid data leakage from the test dataset to the training dataset.

4.5.3. Methods

The generator-adversarial network is a unsupervised learning algorithm based on a combination of two neural networks, a generator and a discriminator. The generator is trained to generate data samples. The discriminator is trained to distinguish correct samples from incorrect ones (Creswell et al, 2018; Wang et al, 2017).

A brief description of the order of the stages of training the generative-adversarial network for this study is shown in Figure 4.5.1.

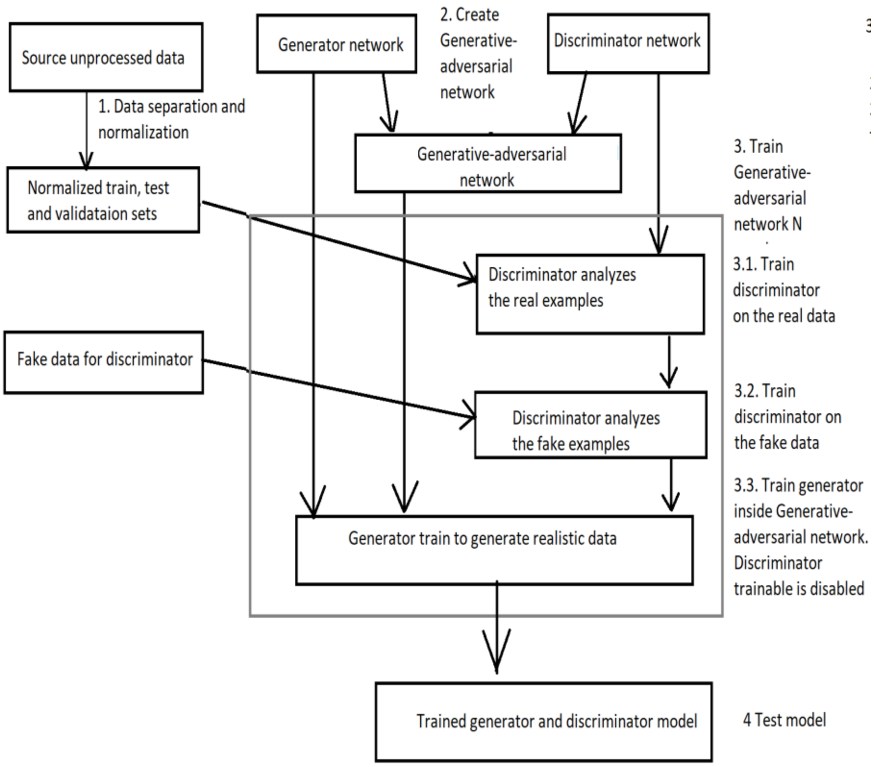


Figure 4.5.1. Stages of training a generative-adversarial network

The discriminator is trained to the task of classifying real and false data generated by the generator. The discriminator is trained on two batches in 1 learning step of the generative-adversarial network. One batch contains real data with a label of real data. The second batch contains generator-generated data with labeled fake data.

Then, in the same training step of the generative-adversarial network, after training the discriminator on the real and generated data, the generator is trained to generate realistic data.

The discriminator's trainability is turned off. The input data and random noise are fed to the input of the generative-adversarial network. The noise is used to generate different values from the same data. The generator within the generation-adversarial network generates the data and sends it immediately to the discriminator. At this point, the generated data is assigned a real data label. The generator is trained to generate more realistic data.

The generator resulting from the training can be used to generate the desired data. The discriminator from the training results can be used to check the generated data for plausibility. For realistic objects generated by the generator, the discriminator score has a higher value. For unrealistic objects, the discriminator returns a low probability value.

The generator-adversarial network is a generator and a discriminator connected in series. The architecture of the generator-adversarial network is shown in Figure 4.5.2

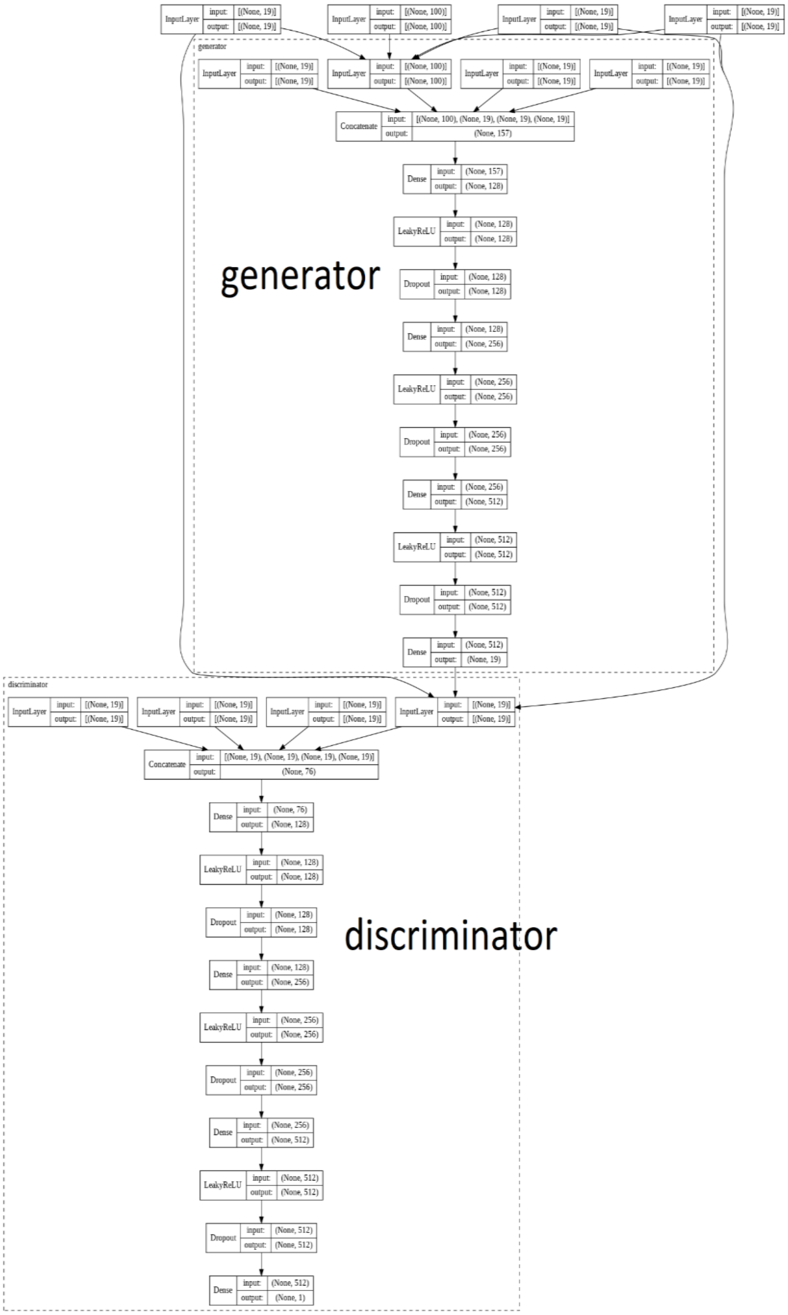


Figure 4.5.2. Architecture of the generative-adversarial

The generator has 4 input layers.

The first input layer receives noise to generate different realistic parameters with the same input values. The noise consists of 100 random parameters generated based on a normal distribution. The noise values are distributed between 0 and 1.

The second input layer is a mask with descriptions of known and unknown parameters. A value of 0 indicates a parameter whose value range is known. A value of 1 indicates a parameter whose value is unimportant or completely unknown. The number of unknown parameters can be up to 10.

On the third input layer comes the minimum acceptable values of the parameters, specified by the user. If a parameter corresponds to an element of mask 1 on the second input layer, there is no information about this parameter and it must be generated. On the third layer this parameter is set to -1. With the parameter normalization option selected earlier in the 0 to 1 range, the value -1 is an unreachable value. This value can be used as a marker for unknown values.

The fourth level of the generator receives the maximum allowable parameter values. If a parameter corresponds to a mask element 1 on the second input layer, there is no information about this parameter and it must be generated. On the fourth layer this parameter is set to -1. With the parameter normalization option selected earlier in the range 0 to 1, the value -1 is an unreachable value. Such a value can be used as a label for unknown values.

If there is no information about the parameter, the value -1 for it must be set simultaneously on the 3 and 4 input layers.

The generator output generates 19 parameters describing the alloy based on the parameters set at the input. The unknown parameters are restored by the generator in the normalized range.

The generator is trained using an average absolute error loss function and Adam's optimizer. The learning rate of the optimizer is 0.001. The batch size is 20.

The discriminator checks the validity of the prediction made by the generator. The discriminator has 4 inputs.

The first input layer of the discriminator receives the prediction made by the generator. These inputs are used to check the realism of the generated data. During discriminator training, this discriminator input receives not only data from the generator, but also realistic data.

The other 3 inputs are fed from the input layers of the generator 2,3,4 without any processing.

Binary cross-entropy loss function and Adam's optimizer are used to train the discriminator. The learning rate of the optimizer is 0.0005. The exponential decay rate for the first moment estimates beta_1 is 0.5. The batch size is 20.

Learning of the discriminator in the generative-adversarial network is disabled. Only the generator will be trained in it. The discriminator is trained separately. To train the generative-adversarial network we use a binary cross-entropic loss function and Adam's optimizer. The learning rate of the optimizer is 0.0005. The exponential decay rate for estimating the first moment beta_1 is 0.5. The batch size is 20.

Model hyperparameters were obtained by HParams Dashboard with Hyperband algorithm (Li et al, 2018).

Figure 4.5.3 shows the change in the error of the generator loss function. The error of the generator decreases i.e. the generator creates more and more realistic objects based on the given parameters.

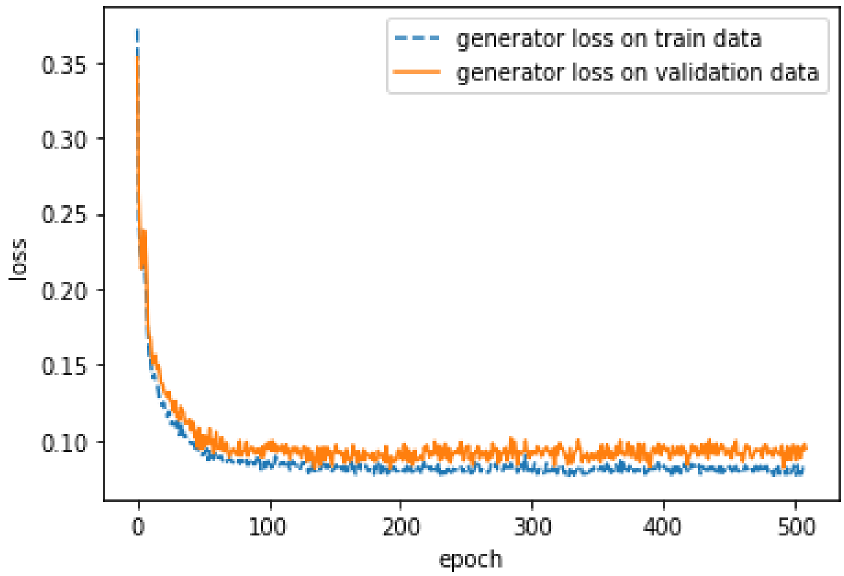


Figure 4.5.3. Generative-adversarial architecture

In Figure 4.5.4 you can see how the generator learns more and more accurately with each new epoch to generate examples with enough realism to bypass the discriminator. It becomes harder for the discriminator to distinguish the generated data from the real data with each new epoch.

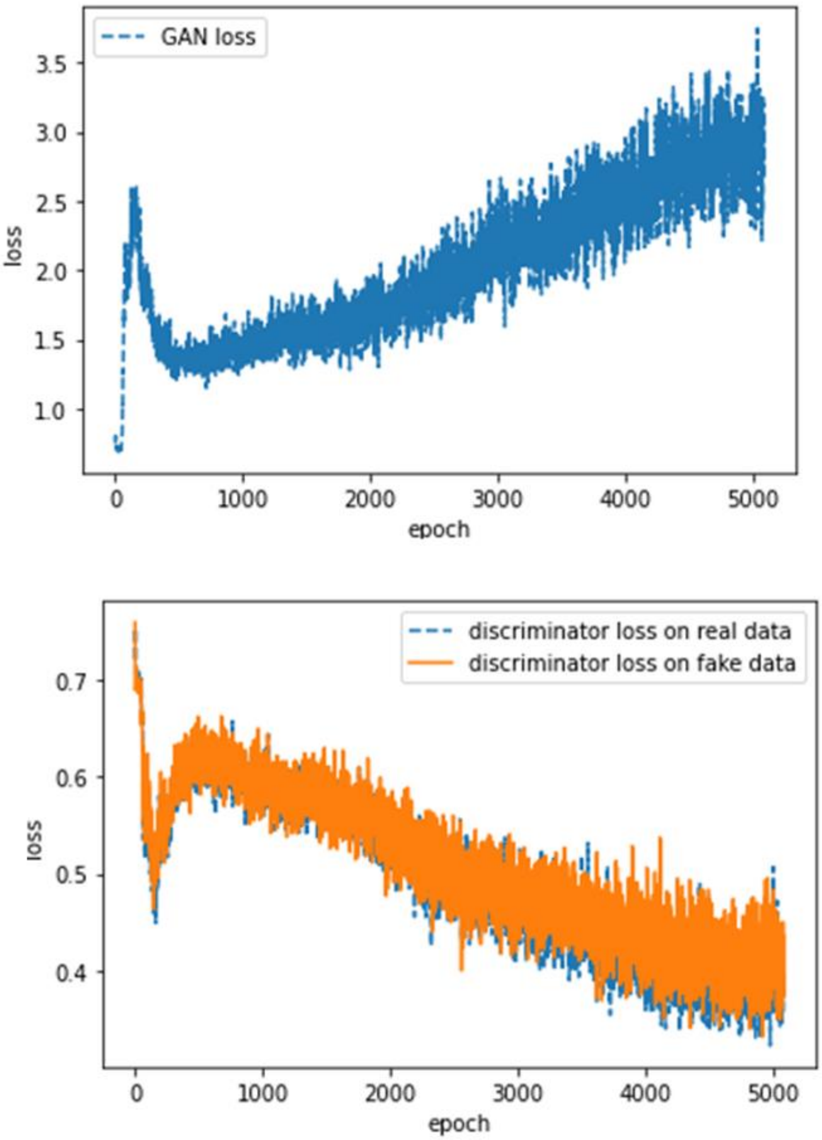


Figure 4.5.4. Learning process of the discriminator. Growth of error (up) and decline in accuracy (down) of the discriminator

The peak value of the error at the initial epochs characterizes the beginning of learning of the generator. The generator at this stage begins to find dependencies in the data. The task of the discriminator changes sharply from classification of real data and random noise to the task of classification of real data and generated data.

4.5.4. Results

With different options for generating weights and neural network architectures, the best results were achieved between 5,000 and 3,000 epochs. The best version is shown below. For the best version, training lasted 990 epochs.

Generator error was checked against the training and validation data every 10 epochs. The generated data were compared to the raw data. The error was calculated based on the loss function used by the average absolute error generator. Figure 6 shows the variation of the generator average absolute error loss function for the training and validation data.

The generator model with the lowest loss function value on the validation data was retained. The discriminator was saved with it.

The generator model with the lowest value of the mean absolute error loss function on the validation data was obtained. The discriminator was saved with the generator at this epoch. The minimum value of the mean absolute error on the validation data is 0.082. Since the parameters were normalized to a range of 0 to 1, this value of the generator loss function can be seen as an accuracy of 91.8% relative to the real value of the calculated parameters.

Using a lower learning rate increased the final value of the oscillator loss function. Presumably, the training of the model weights stopped at a local minimum. Training the model at a higher rate reduced the probability of successful convergence of the oscillator and model discriminator.

The discriminator's performance can be used to assess the degree of realism of the proposed material. The value returned by the discriminator increases as the realism of the material proposed by the generator increases.

Neural networks can also be trained to predict the missing material property based on what is already known. If the generator is trained well enough to generate realistic examples, the error in predicting the missing parameter will not exceed the sum of the error of the generator and the trained model on the predicted parameter.

Having trained the models to predict the missing parameter on real data, such models can be used to evaluate the realism of the generated materials.

By analyzing each parameter with a separate neural network and predicting its value based on the other known parameters, you can get an error value for each parameter. If you add up these values and look at the discriminator value, you can find out how realistic the material was generated. Figure 4.5.5 shows the correlation of the discriminator value predicting the realistic material along the X axis and the calculated error for the generated alloy under given conditions along the Y axis.

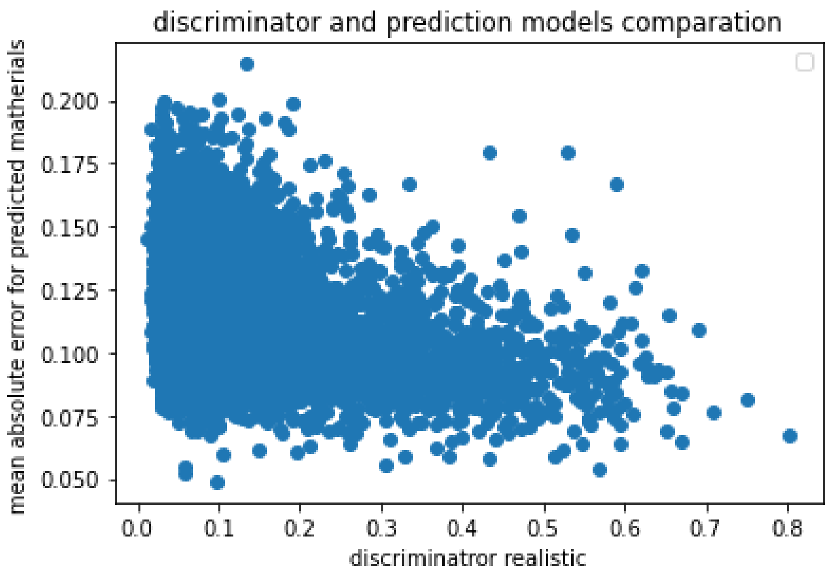


Figure 4.5.5. Correlation of discriminator prediction and neural network error

The generator generated 200000 variants of material with different levels of realism. You can see that most of the materials have a low level of realism. The higher is the discriminator value, the higher is the material realism. Generated materials with discriminator value less than 0.5 have less than 50% probability to have characteristics close to the generated ones. Such materials are of no interest.

Of particular interest are materials with high realism and low error in the lower right part of Figure 5.5. These material values can exhibit the desired properties and parameters with the highest probability. However, they need to be further constrained by tight limits on the required parameters with a small tolerance on all analyzed parameters. The generator is a neural network and therefore never returns a perfectly accurate value. For example, if the upper and lower bounds of the parameter range are 0, the neural network will not return 0, but a value close to 0.

4.5.5. Conclusions

This subsection "Neural network for generation of composition and parameters of metal alloys based on specified range of known and unknown parameters" describes the development of a generative-adversarial neural network for generating metal alloy connections with given parameters.

As a result, the generative-adversarial neural network has been trained. It is now capable of generating composition and properties of alloys with an average absolute error of 0.082 units relative to the normalized between 0 and 1 range of validation data parameters, i.e., with an accuracy of 91.8% relative to the real value.

Generation of objects with required parameters based on known parameters can be applied in various fields. This approach can be used not only for the generation of metal alloys, but also for other data. For example, materials science can train a model on data from related fields and then combine the trained generators and discriminators into ensembles. Such an approach will be able to describe the physical effects of materials in a versatile way and avoid the problem of stopping learning at the local minimum of the feature space.

Adding additional features to the data can improve the quality of developed models. For example, such features can be parameters of chemical elements from periodic table of elements.

In the future, it is planned to train the considered approach for related areas of materials science. It is also planned to investigate the possibilities of combining such trained models into ensembles.

4.6 Case study 2. A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications. Conclusion

In this subsection the possibilities of using neural networks for materials analysis and prediction of their parameters have been considered.

Various approaches to the analysis of structure, composition and properties of materials and prediction of unknown target parameters on the example of different data in different areas in subsections 4.2, 4.3, 4.4, 4.5 have been studied. These subsections show the development of the proposed idea.

The use of neural networks has been shown to be viable only when appropriate preliminary data analysis is applied. Depending on the quality and quantity of data, different neural network architectures and data processing algorithms must be used.

Main obtained result points:

1. An algorithm for processing and representing composition and material properties data to train neural networks for each of the cases considered. Different approaches to the analysis and processing of material parameters were demonstrated.

2. Development and training algorithms and trained neural network model for prediction of critical superconductivity temperature for superconducting materials based on their chemical formula. This work has been performed and is described in detail in subsection 4.2. This subsection focuses on exploring the possibility of using additional parameters of chemical elements from the periodic table of elements to train neural networks. However, this subsection has the disadvantage that it is not possible to predict the composition of the material based on the required parameters. Algorithms that allow generating are considered in further sections.

3. Development and training algorithms and a trained neural network model for predicting the reduced glass transition temperature of metal alloys based on a neural network. This work is performed and described in detail in subsection 4.3. This subsection explores the possibilities of predicting target material parameters without using additional parameters from the periodic table and additional balancing of the data. The possibility of staggered learning in several stages using additional output parameters under data

constraints is also studied. The considered approaches allow working with limited data sets and predicting material parameters. However, this approach does not allow us to generate material composition based on the required material parameters.

4. Development and training algorithms and a trained neural network model for predicting the composition of paper-like materials based on the specified chemical and physical parameters. These works are performed and described in detail in subsection 4.4. This subsection focuses on working with ultra-small amounts of data and predicting material composition based on its parameters. A material can have a variety of parameters. These parameters can have a huge number of variations. Using these parameters as input data for a neural network made it possible to describe the composition of the material with a small amount of data. But the inverse task could not be solved for the investigated data because 5 parameters searching material composition was not enough to learn prediction of 14 physical parameters of material with very little data. Also, the problem was weakened by the fact that the material composition has well-defined physical properties, but several materials can have the same physical properties. On a small amount of data, this problem is pronounced, but a solution to this problem is not possible. This issue is solved in the following subsection

5. Development and learning algorithms and a trained neural network model for generating metal alloy composition and parameters based on a specified range of known and unknown parameters. This work has been done and is described in detail in subsection 4.5. This subsection discusses an approach to predicting material properties on the basis of generative-adversarial neural networks for a limited amount of data. Composition and property parameters are normalized and predicted jointly. In this way, data can be reconstructed for a number of unknown parameters. A method for assessing the reliability of the reduced parameter prediction is also proposed. This approach allows us to solve the problem of a small amount of data. The problem of comparing the material properties and the set of its possible compositions is also solved by using random noise and an algorithm for assessing the reliability of the prediction.

Case Study 2, "A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications" is linked to items 8 and 9 on the list for the “good health and well-being” goal of the Sustainable Development Goals 2030.

This section 4 describes process and results of research of using neural networks for materials analysis tasks can reduce the final cost of materials development. It can simplify the manufacturing process for the creation of the materials with different target properties. This approach will make widely available materials with different properties for different science, social, industrial, and medical areas, which corresponds to the point 8 of the list for the "good health and well-being" goal of the Sustainable Development Goals 2030. Also, this approach can help in the development of safe environmentally friendly materials and means to recycle biology and chemically active substances, which corresponds to the goal 9 of the list for the "good health and well-being" goal of the Sustainable Development Goals 2030.

Also, case study 2 "A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications" is related to points 4, 5 and 7 in objectives list for the industry, innovation and infrastructure goal of the Sustainable Development Goals 2030.

The application of neural networks to analyze the properties, composition and parameters of materials can allow to automate, simplify, improve the process of finding, studying, creating materials with given parameters and structure based on known parameters and from given components.

This approach may allow the development of materials with more pronounced required parameters, solve environmental and economic production by simplifying and reducing the cost of production chains. These arguments allow to realize the point 4 of Industry, innovation and infrastructure goal objectives. Section 4 describe algorithms for different tasks: for prediction physical and chemistry properties based on the composition; prediction of the composition based on physical and chemistry properties; generation of composition and parameters of material based on range of known and unknown parameters.

Making it easier to develop materials using artificial intelligence techniques will allow more people to develop new materials. The cost of testing new materials will decrease by reducing the cost, complexity, and number of experiments required to achieve the desired material parameters. The emergence of new materials will create new industries and process chains. These arguments allow to realize the points 5 and 7 of Industry, innovation, and infrastructure goal objectives.

The methods presented in Section 4 for using artificial intelligence for materials analysis show that this approach can be used for various types of materials analysis tasks. The use of artificial intelligence can lower the entry threshold into this field and increase the number of researchers and private research companies in this field.

5 CONCLUSIONS

This section presents the most significant conclusions drawn after the development of this work. These conclusions are presented after completing the tasks outlined in the Introduction chapter.

This dissertation began by exploring the potential of machine learning to track and evaluate the progression and treatment of multiple sclerosis by tracking the position of the fingers of the hand and measuring fingers mobility. This topic was named “deep learning techniques applied to predict and measure finger movement in patients with multiple sclerosis” and was chosen as the first case study.

The topic explored the application of neural networks in medical area, developed algorithms to handle the data, explored algorithms for training neural networks and the principles of convolutional neural networks. Algorithms of data augmentation allowing to hide certain features in data, possibilities of images segmentation were developed and tested.

Further, in the process of research, the idea and understanding appeared that successful development of different spheres of human activity, development in social, medical, scientific or technical fields required not only algorithmic, but also material basis.

The question of predicting material properties on the basis of its chemical formula was studied, using the temperature prediction of superconductivity of materials as an example. Neural networks have shown that they are able to analyze the physical parameters of materials. This area was chosen as the second case study and was named “a deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications”. Next, the question of working with a limited amount of data and the issue of data non-uniformity in glass transition temperature prediction was investigated.

As a result, enough results were obtained to move on to the task of predicting material composition based on its properties using paper-like materials as an example. In this work, there was an extremely limited amount of data because measuring the structure of materials is more difficult than changing their properties. The peculiarities of the influence of initial values of weights on the results of neural network training with minimal data size were studied.

As a result, a generative-adversarial network for predicting material parameters based on incomplete data about them was developed. Metal alloy parameters were used as input data, but a similar approach can also be used to predict parameters of any other materials. For the generative-adversarial neural network, an algorithm was developed to evaluate the quality of the generated example based on pre-trained neural networks and a discriminator.

The data had different representations at different stages. Depending on the type of representation of the materials data, different options for data processing and analysis are given.

A variety of data sources were used, as each demonstrated the maximum level of difficulty manifested in the previous stage of the work. The use of different data sources allowed a versatile study of the issues that arose and the adaptation of the developed algorithms to different conditions. The data processing algorithms and neural network training approaches developed in this thesis can be used in various fields, offering different solutions to problematic situations that may arise in the analysis of material properties and neural network training.

5.1 Objectives, hypotheses and Research Questions

This dissertation work has made several contributions to artificial intelligence methods and biomedicine: algorithms for data collection, processing and analysis have been studied, neural networks of different architectures have been developed and trained for application tasks on different data, reliable results have been obtained using typical validation metrics for each application. The completion of the various phases and contributions of this dissertation was made possible by achieving the goals presented in Section 1.1.

Two case studies with different research questions and specific objectives were selected:

Case study 1. Deep Learning Techniques applied to predict and measure finger movement for the patients with multiple sclerosis

Case study 2. A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications

The following research questions were selected and answered for case study 1 “Deep Learning Techniques applied to predict and measure finger movement for the patients with multiple sclerosis” in this dissertation:

RQ 1. Deep learning algorithms can automatically measure with high accuracy the angles value of the hand in one image. This research question has been explored in section 3. The obtained results and comparation with different methods are presented in the table 3.5.1. Ability of the deep learning algorithms for the automatically measure with high accuracy the angles value of the hand in one image has been explored and confirmed.

RQ 2. The use of non-standard ways of data augmentation allows to expand the expand the possibilities of applying the neural network and increase its stability. This research question has been explored in section 3. A data augmentation algorithm was developed to hide the features of the data in the images. This algorithm can be used to train a neural network to find hidden features in the data. However, this algorithm can only be used in ensemble with an algorithm trained on raw images. The use of this data augmentation algorithm in real-time applications is difficult because of performance issues.

Based on the hypothesis and the research questions posed, the goal of this dissertation for this case study is to develop a system for predicting and measuring finger movements in patients with multiple sclerosis based on deep learning and computer vision algorithms. This aim has been reached. Results are presented in the table 3.5.1.

To reach the main aim of this dissertation, the following specific objectives SO need to be fulfilled.

SO 1. Determine the possibilities of different computer vision algorithms to solve the problem of measuring the joint angles of fingers. This specific objective has been explored in sections 3.1 and 3.2. Various resources have been analyzed from different sides.

SO 2. Collect data for training the computer vision system to measure the angles of the joint angles of fingers. This specific objective has been explored in section 3.3. The simulation and high data augmentation approach has been selected. This approach was chosen because of the low amount of available data suitable for this case study and the unavailability to use additional sensors.

SO 3. Develop and software implementation of computer vision algorithm for measuring the joint angles of fingers. Training neural network model. This specific objective has been explored in section 3.4. A special learning algorithm was developed for gradually increasing complexity of the data. Training was performed in several stages with gradual replacement of labeled images by unlabeled images with augmentation.

SO 4. Create of the test version of the program. Debugging and testing the developed algorithm. To draw conclusions about the system quality. This specific objective has been explored in sections 3.4 and 3.5. Several versions of the neural network were developed and tested: with background subtraction, with additional hiding of data features.

SO 5. Create of the final real time implementation of the program based on the developed algorithm. This specific objective has been explored in sections 3.6 and 3.5. The use of ensemble based on developed data augmentation algorithm in real-time applications is difficult because of performance issues. The obtained results and comparation with different methods are presented in the table 3.5.1.

Case study 1 is related to the “Good health and well-being” section of the Sustainable Development Goals 2030.

The developed in this case study 1 system can operate in real time, has low cost, is easy to service, and can monitor the development of hand mobility limitations without the use of expensive sensors. The high availability of the system is consistent with items 8 “Achieve universal health coverage, including financial risk protection, access to quality essential health services and access to safe, effective, quality and affordable essential medicines and vaccines for all” and 13 “Strengthen the capacity of all countries, in particular developing countries, for early warning, risk reduction, and management of national and global health risks” of the “Good health and well-being” section of the Sustainable Development Goals 2030 list.

Early diagnosis of multiple sclerosis, rapid response to the manifestation of its symptoms and tracking the process of treatment can significantly improve the quality of life and life expectancy of people with this disease. The use of the developed system can help to achieve the item 4 “By 2030, reduce premature deaths from non-communicable diseases by one-third through prevention and treatment and promote mental health and well-being” of the “Good health and well-being” section of the Sustainable Development Goals 2030 list.

Detailed summary with the results and developed algorithms of the case study 1 is presented in section 3.7.

The following research questions were selected and answered for case study 2 “A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications” in this dissertation:

RQ 1. Neural networks can predict the properties of materials based on their chemical composition. This research question has been explored in sections 4.2 and 4.3. The obtained results and comparation with different methods are presented in the table 4.2.5, sections 4.3.3 and 4.3.4. Ability of the neural networks for the prediction of the properties of materials based on their chemical composition has been explored and confirmed.

RQ 2. Neural networks can predict the composition of materials based on their physical and chemistry properties. This research question has been explored in section 4.4. The obtained results and comparation with different methods are presented in the section 4.4.5. Ability of the neural networks for the prediction of the composition of materials based on their physical and chemistry properties has been explored and confirmed.

RQ 3. Neural networks can be used for generation of the material properties and composition description and evaluation of the generated descriptions. This research question has been explored in section 4.5. The obtained results and comparation with different methods are presented in the section 4.5.4.

Ability of the neural networks for generation of the material properties and composition description and evaluation of the generated descriptions has been explored and confirmed

Based on the hypothesis and the research questions posed, the goal of this dissertation for this case study is to develop a system for predicting and measuring properties and composition of materials based on incomplete information about the material. This aim has been reached in section 4.5. This is result of each previous step.

To reach the main aim of this dissertation, the following specific objectives need to be fulfilled.

SO 1. Analyze various approaches to materials analysis. This specific objective has been explored in sections 4.2.1, 4.2.2, 4.3.1, 4.3.2, 4.4.1, 4.4.2, 4.5.1, 4.5.2. Various resources have been analyzed from different sides.

SO 2. Collect and prepare data for analyzing the properties of materials. This specific objective has been explored in sections 4.2.2, 4.3.2, 4.4.2, 4.5.2. A variety of data sources were used, as each demonstrated the maximum level of difficulty manifested in the previous stage of the work. The use of different data sources allowed a versatile study of the issues that arose and the adaptation of the developed algorithms to different conditions.

SO 3. Develop an algorithm and train a neural network to predict target parameters of materials. This specific objective has been explored in sections 4.2.3, 4.3.3, 4.4.3, 4.5.3. The data processing algorithms and neural network training approaches developed in this thesis can be used in various fields, offering different solutions to problematic situations that may arise in the analysis of material properties and neural network training.

SO 4. Develop a program to use the developed trained neural network model. This specific objective has been explored in sections 4.2.4, 4.3.4, 4.4.4, 4.5.4. For each type of task a different data processing algorithm and a different learning algorithm were used.

SO 5. Test and debug the developed program.

This case study was conducted in several steps. This specific objective has been explored in sections 4.2.4, 4.3.4, 4.4.4, 4.5.4, 4.2.5, 4.3.5, 4.4.5, 4.5.5, 4.6 . The best obtained results are shown in the conclusion subsections of the corresponding sections.

In step 1, neural network prediction of critical superconductivity temperature for materials based on their chemical formula was considered. It was described in section 4.2.

Step 2 considered the prediction of the reduced glass transition temperature of metal alloys based on a neural network. It was described in section 4.3.

Step 3 considered the prediction of material composition based on the required physical parameters for cellulosic materials. It was described in section 4.4.

Step 4 considered the use of generative-adversarial networks to predict the properties and composition of metal alloys based on incomplete material information with an acceptable range of predicted parameters. It was described in section 4.5.

Case Study 2, "A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications" is linked to items 8 and 9 on the list for the “good health and well-being” goal of the Sustainable Development Goals 2030.

This case study describes process and results of research of using neural networks for materials analysis tasks can reduce the final cost of materials development. It can simplify the manufacturing process for the creation of the materials with different target properties. This approach will make widely available materials with different properties for different science, social, industrial, and medical areas, which corresponds to the point 8 “Achieve universal health coverage, including financial risk protection, access to quality essential health services and access to safe, effective, quality and affordable essential medicines and vaccines for all” of the list for the “good health and well-being” goal of the Sustainable Development Goals 2030. Also, this approach can help in the development of safe environmentally friendly materials and means to recycle biology and chemically active substances, which corresponds to the goal 9 “By 2030, significantly reduce the number of deaths and illnesses from hazardous chemicals and air, water, and soil pollution and contamination” of the list for the “good health and well-being” goal of the Sustainable Development Goals 2030.

Also, case study 2 “A deep learning approach to predict the properties and composition of materials with potential capabilities in biomedical applications” is related to points 4, 5 and 7 in objectives list for the industry, innovation and infrastructure goal of the Sustainable Development Goals 2030.

The application of neural networks to analyze the properties, composition and parameters of materials can allow to automate, simplify, improve the process of finding, studying, creating materials with given parameters and structure based on known parameters and from given components.

This approach may allow the development of materials with more pronounced required parameters, solve environmental and economic production by simplifying and reducing the cost of production chains. These arguments allow to realize the point 4 “By 2030, modernize infrastructure and retool industry to make it sustainable, with increased resource efficiency and greater adoption of clean and environmentally sound technologies and industrial processes, with all countries acting according to their capabilities” of Industry, innovation and infrastructure goal objectives. This case study 2 describe algorithms for different tasks: for prediction physical and chemistry properties based on the composition; prediction of the composition based on physical and chemistry properties; generation of composition and parameters of material based on range of known and unknown parameters.

Making it easier to develop materials using artificial intelligence techniques will allow more people to develop new materials. The cost of testing new materials will decrease by reducing the cost, complexity, and number of experiments required to achieve the desired material parameters. The emergence of new materials will create new industries and process chains. These arguments allow to realize the points 5 “Intensify research, improve the technological capabilities of industrial sectors in all countries, especially in developing countries, including, by 2030, encouraging innovation and substantially increasing the number of research and development workers by 1 million, as well as public and private spending on research and development” and 7 “Support the development of domestic technology, research and innovation in developing countries, including by providing an enabling policy environment for, inter alia, industrial diversification and value addition to commodities

8. Significantly increase access to information and communications technology and strive to provide universal and affordable access to the Internet in the least developed countries by 2030” of Industry, innovation, and infrastructure goal objectives.

The methods presented in this case study 2 for using artificial intelligence for materials analysis show that this approach can be used for various types of materials analysis tasks. The use of artificial intelligence can lower the entry threshold into this field and increase the number of researchers and private research companies in this field.

Detailed summary with the results and developed algorithms of the case study 2 is presented in section 4.6.

Listed below are the hypotheses, specific goals and their corresponding research questions that have been answered in this dissertation.

The following main hypotheses are highlighted and studied in the process of research:

1. Artificial intelligence methods have a wide range of applications and can solve different social challenges in biomedical field. This hypothesis was confirmed in sections 3 and 4 of the dissertation. The applicability of neural networks for different types of tasks was demonstrated. The best correct results can be obtained with appropriate correct processing and analysis of the original data. The research was conducted with the Sustainable Development Goals 2030 in view. Sustainable Development Goals 2030 aims to solve various types of challenges, increase level of human live, social challenges. The considered research cases correspond to goals 3 “Good health and well-being” and 9 “Industry, innovation, and infrastructure” of the plan, designed to solve social challenges and raising living standards from a medical and industrial point of view.

2. Deep learning algorithms can automatically measure with high accuracy the angles value of the hand in one image. This hypothesis was confirmed in section 3 of the dissertation. In this case, a neural network using convolutional layers based on MobileNetV2 architecture was applied.

3. Neural networks can be used for the material analysis for the prediction of the properties of materials based on their composition and prediction the materials composition based on the material composition. This hypothesis was confirmed in section 4 of the dissertation in 4 steps. In step 1, neural network prediction of critical superconductivity temperature for materials based on their chemical formula was considered. Step 2 considered the prediction of the reduced glass transition temperature of metal alloys based on a neural network. Step 3 considered the prediction of material composition based on the required physical parameters for cellulosic materials. Step 4 considered the use of generative-adversarial networks to predict the properties and composition of metal alloys based on incomplete material information with an acceptable range of predicted parameters.

5.2 Scientific contribution

The following presents a complete relation of the different publications that took part throughout this research work. This dissertation is structured as a monography. Two of the studies are published, and one is conference article in publishing process, all in international journals with impact factors. Other contributions have also been made to the scientific community in the shape of communications to different international conferences, which are summarized below.

5.2.1 Articles in international journals with impact factor

The articles detailed in this section are the ones that compound this Ph.D. dissertation. Two of them have already been accepted and published in international journals, while the last one is currently under review.

Table 5.2.1. Publication I - Details of the publication.

Title	Deep Learning Techniques Applied to Predict and Measure Finger Movement in Patients with Multiple Sclerosis		
Authors	Dmitrii Viatkin, Begonya Garcia-Zapirain Amaia Méndez Zorrilla		
Journal	*Applied science, MDPI*		
Publication	01 April 2021		
Impact	2.68	Quartile	Q2
DOI	https://doi.org/10.3390/app11073137		

Table 5.2.2. Publication II - Details of the publication.

Title	Deep Learning Approach for Prediction of Critical		
Authors	Dmitrii Viatkin, Begonya Garcia-Zapirain Amaia Méndez Zorrilla Maxim Zakharov		
Journal	*Information*		
Publication	27 October 2021		
Impact	3.51	Quartile	Q2
DOI	https://doi.org/10.3389/fmats.2021.714752		

5.2.2 Communications in international conferences

These articles, posters, and communications have been produced during the Ph.D. research. All the communications in this section are related to the topic and research field of this dissertation.

Table 5.2.3. Publication III - Conference publication.

Title	Prediction of reduced glass transition temperature of metallic alloys based on a neural network
Authors	Dmitrii Viatkin Maxim Zakharov Dmitrii Zhuro
Conference	III International Conference on Metrological Support of Innovative Technologies (ICMSIT-III-2022)
Year	03-06 March 2022 **Location** Russia
Publisher	Journal of Physics: Conference Series (JPCS)

Table 5.2.4. Publication IV - Conference publication.

Title Calculation of northern hemisphere sea ice area using recurrent neural networks			
Authors	Dmitrii Viatkin Maxim Zakharov Dmitrii Zhuro Svetlana Malysheva		
Conference	Fundamental and Applied Scientific Research in the Development of Agriculture in the Far East (AFE 2021)		
Publisher	IOP Conference Series: Earth		
Year	20-21 June 2021	**Location**	Russia
DOI	10.1088/1755-		

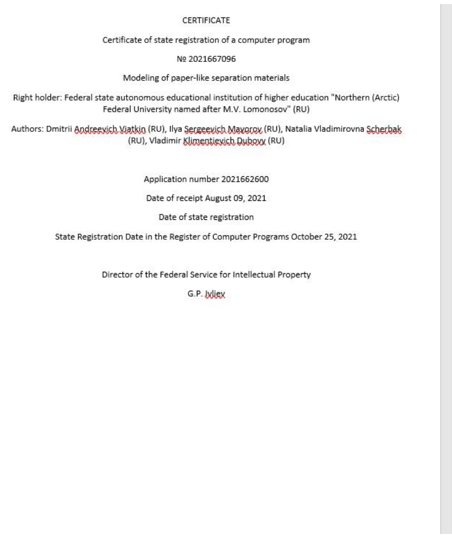
Table 5.2.5. Publication V - Conference publication.

Title DeepDream algorithm for data augmentation in a neural network ensemble applied to multiclass image classification		
Authors	Dmitrii Viatkin Begonya Garcia-Zapirain Amaia Méndez Zorrilla	
Conference	14th Asian Conference on Intelligent Information and Database Systems (ACIIDS 2022)	
Year	28-30 November **Location**	Vietnam

5.2.3 Patent registration certificates

In the process of research, a patent was obtained, confirming the registration of the algorithm developed during the research.

In October 2021, a patent was obtained in Russia for the use of a variation of the algorithm from section 4.4 to predict the parameters of pulp and paper materials. Original scan of certificate and translation is shown below.



CERTIFICATE

Certificate of state registration of a computer program

№ 2021667096

Modeling of paper-like separation materials

Right holder: Federal state autonomous educational institution of higher education "Northern (Arctic) Federal University named after M.V. Lomonosov" (RU)

Authors: Dmitrii Andreevich Viatkin (RU), Ilya Sergeevich Mayorov (RU), Natalia Vladimirovna Scherbak (RU), Vladimir Klimentievich Dubovy (RU)

Application number 2021662600

Date of receipt August 09, 2021

Date of state registration

State Registration Date in the Register of Computer Programs October 25, 2021

Director of the Federal Service for Intellectual Property

G.P. Ivliev

РОССИЙСКАЯ ФЕДЕРАЦИЯ

СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2021667096

Моделирование бумагоподобных материалов сепарационного назначения

Правообладатель: *Федеральное государственное автономное образовательное учреждение высшего образования «Северный (Арктический) федеральный университет имени М.В. Ломоносова» (RU)*

Авторы: *Вяткин Дмитрий Андреевич (RU), Майоров Илья Сергеевич (RU), Щербак Наталья Владимировна (RU), Дубовый Владимир Климентьевич (RU)*

Заявка № 2021662600

Дата поступления 09 августа 2021 г.

Дата государственной регистрации в Реестре программ для ЭВМ *25 октября 2021 г.*

Руководитель Федеральной службы по интеллектуальной собственности

Г.П. Ивлиев

Figure 5.2.1. Patent for the developed algorithm

Patent has number 2021667096. Full information about the certificate can be is available on the official page https://www.fips.ru/en/.

5.3 Limitations and Recommendations for Future Research

Many aspects of the research have been considered in the design and implementation of this dissertation. However, there are some limitations that are worth pointing out. Each of these limitations and future work are presented in their respective chapters, but a broad overview is also presented here.

A lot of attention is given to the question of data collection, analysis, and processing. In the process, correct and balanced processing of the data is an important aspect for a successful correct result. This point is especially strong in cases of small data sets and simulated datasets. Highlighting additional features of the data can help solve this problem. This points were explained in case study 1 and case study 2 in sections 3 and 4.

The datasets from different fields are considered in the dissertation. This choice of datasets is due to the desire to explore the different facets of data analysis that can be encountered when working with real data. These datasets are chosen because they can help in the research of data analysis features on small sets of unbalanced noisy data.

In the case study 2, when analyzing materials, shown that the focus of the dissertation is shifted toward a general analysis of the structure and properties of materials, not just the biomedical application part. This is done to eliminate the subjectivity of perception in analyzing the data and obtaining a correct numerical evaluation of the results obtained. As a result of this approach, an approach for generating metal alloys based on incomplete data was developed in the section 4.5. Since the data analysis and neural network training did not focus on the specifics of the data application, the developed approach can be used for biomedical applications or other areas of materials development.

The next step in the research could be the use of additional information about the internal structure of the material, the position of atoms and molecules in it. Organic materials are actively used in medicine. Their chemical composition may be the same, but their properties depend on the arrangement of atoms in the material. Modernization of the approach developed in the case study 2, considering the structure of the material may allow the design of target materials.

The development of algorithms to produce different materials can also be considered a development of the ideas of this dissertation. This may make it possible to obtain materials, the properties of which now do not look realistic according to the new technological processes.

REFERENCES

Claps, G. G., Berntsson Svensson, R., & Aurum, A. (2015). On the journey to continuous deployment: Technical and social challenges along the way. Information and Software Technology, 57, 21–31.

Cath, C. (2018). Governing artificial intelligence: ethical, legal and technical opportunities and challenges. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 376(2133), 20180080.

D'Oca, S., Ferrante, A., Ferrer, C., Pernetti, R., Gralka, A., Sebastian, R., & op 't Veld, P. (2018). Technical, Financial, and Social Barriers and Challenges in Deep Building Renovation: Integration of Lessons Learned from the H2020 Cluster Projects. Buildings, 8(12), 174.

Gerke, S., Minssen, T., & Cohen, G. (2020). Ethical and legal challenges of artificial intelligence-driven healthcare. Artificial Intelligence in Healthcare, 295–336.

Glikson, E., & Woolley, A. W. (2020). Human Trust in Artificial Intelligence: Review of Empirical Research. Academy of Management Annals.

Price, W. N., Gerke, S., & Cohen, I. G. (2019). Potential Liability for Physicians Using Artificial Intelligence. JAMA, 322(18), 1765.

Dignum, V. (2018). Ethics in artificial intelligence: introduction to the special issue. Ethics and Information Technology, 20(1), 1–3.

Mutascu, M. (2021). Artificial intelligence and unemployment: New insights. Economic Analysis and Policy, 69, 653–667.

Prentice, C., Dominique Lopes, S., & Wang, X. (2019). Emotional intelligence or artificial intelligence– an employee perspective. Journal of Hospitality Marketing & Management, 1–27.

Amann, J., Blasimme, A., Vayena, E., Frey, D., & Madai, V. I. (2020). Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. BMC Medical Informatics and Decision Making, 20(1).

Hamet, P., & Tremblay, J. (2017). Artificial intelligence in medicine. Metabolism, 69, S36–S40.

Keskinbora, K. H. (2019). Medical ethics considerations on artificial intelligence. Journal of Clinical Neuroscience.

Thomas M., Daniele T., (2022) Path dependence and stagnation in a classical growth model. Cambridge Journal of Economics, Volume 46, Issue 1, January 2022, Pages 195–218,

Alexis M. (2021) From the post-industrial prophecy to the de-industrial nightmare: Stagnation, the manufacturing fetish and the limits of capitalist wealth. Competition & Change.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1798–1828.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444.

Anders, S., & Huber, W. (2010). Differential expression analysis for sequence count data. Genome Biology, 11(10), R106.

Boyd, S. (2010). Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. Foundations and Trends® in Machine Learning, 3(1), 1–122.

Martinez, A. M., & Kak, A. C. (2001). PCA versus LDA. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2), 228–233.

Tin Kam Ho. (1998). The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8), 832–844.

Kanungo, T., Mount, D. M., Netanyahu, N. S., Piatko, C. D., Silverman, R., & Wu, A. Y. (2002). An efficient k-means clustering algorithm: analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), 881–892.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM. ACM Transactions on Intelligent Systems and Technology, 2(3), 1–27.

Cortes, C., & Vapnik, V. (1995). Machine Learning, 20(3), 273–297.

Friedman, J. H. (2001). machine. The Annals of Statistics, 29(5), 1189–1232.

Dietterich, T. G. (2000). Ensemble Methods in Machine Learning. Lecture Notes in Computer Science, 1–15.

Bengio, Y. (2009). Learning Deep Architectures for AI. Foundations and Trends® in Machine Learning, 2(1), 1–127

Mallat, S. (2016). Understanding deep convolutional networks. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150203.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A Review of Recurrent Neural

Networks: LSTM Cells and Network Architectures. Neural Computation, 1–36.

Baek, E., Kwon, D., & Kim, J. (2020). A Multi-Neural Network Acceleration Architecture. 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA).

Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. 2017 International Conference on Engineering and Technology (ICET).

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. IEEE Transactions on Neural Networks and Learning Systems, 1–21.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.

Che, Z., Purushotham, S., Cho, K., Sontag, D., & Liu, Y. (2018). Recurrent Neural Networks for Multivariate Time Series with Missing Values. Scientific Reports, 8(1).

Ullah, A., Ahmad, J., Muhammad, K., Sajjad, M., & Baik, S. W. (2018). Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features. IEEE Access, 6, 1155–1166.

Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T. (2017). Adversarial Discriminative Domain Adaptation. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Cosh, A.; Carslaw, H. Multiple sclerosis: Symptoms and diagnosis. InnovAiT 2014, 7, 651–657

Chen, F.; Deng, J.; Pang, Z.; Nejad, M.B.; Yang, H.; Yang, G. Finger Angle-Based Hand Gesture Recognition for Smart Infrastructure Using Wearable Wrist-Worn Camera. Appl. Sci. 2018, 8, 369,

Zhou, Y.; Jiang, G.; Lin, Y. A novel finger and hand pose estimation technique for real-time hand gesture recognition. Pattern Recognit. 2016, 49, 102–114

Kim, J.S.; Kim, B.K.; Jang, M.; Kang, K.; Kim, D.E.; Ju, B.-K.; Kim, J. Wearable Hand Module and Real-Time Tracking Algorithms for Measuring Finger Joint Angles of Different Hand Sizes with High Accuracy Using FBG Strain Sensor. Sensors 2020, 20, 1921

Lu, S.; Chen, D.; Liu, C.; Jiang, Y.; Wang, M. A 3-D finger motion measurement system via soft strain sensors for hand rehabilitation. Sens. Actuators A Phys. 2018, 285, 700–711

Tran, D.P.; Morita, D.; Sato, N.; Morita, Y.; Takekawa, M. Improvement of non-invasive semi-automatic test device for measurement of finger joints range of motion: Reduction in burden on therapist. In Proceedings of the 2016 16th International Conference on Control, Automation and Systems (ICCAS), Gyeongju, Korea, 16–19 October 2016; pp. 423–427.

Mayer, S.; Mayer, M.; Henze, N. Feasibility analysis of detecting the finger orientation with depth cameras. In Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services—MobileHCI '17, Vienna, Austria, 4-7 September 2017.

Maruyama, Y.; Kono, Y. Estimating Finger Postures by Attaching an Omnidirectional Camera to the Center of a User's Palm. In Proceedings of the 2018 International Conference on Advanced Visual Interfaces, Riva del Sole, Castiglione della Pescaia, Grosseto, Italy, 29 May–1 June 2018.

Park, K.; Kim, S.; Yoon, Y.; Kim, T.-K.; Lee, G. DeepFisheye: Near-surface multi-finger tracking technology using fisheye camera. In Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology, Virtual Event USA, 20-23 October 2020; Volume 3415818, pp. 1132–1146.

Kılıboz, N.Ç.; Güdükbay, U. A hand gesture recognition technique for human–computer interaction. J. Vis. Commun. Image Represent. 2015, 28, 97–104

Fan, Z.; Valentin, B.; Andrey, V.; Andrei, T.; George, S.; Chuo-Ling, C.; Matthias, G. MediaPipe Hands: On-device Real-time Hand Tracking. arXiv 2020 arXiv:2006.10214.

Kim, K.K.; Ha, I.; Kim, M.; Choi, J.; Won, P.; Jo, S.; Ko, S.H. A deep-learned skin sensor decoding the epicentral human motions. Nat. Commun. 2020, 11, 2149.

Ashiquzzaman, A.; Lee, H.; Kim, K.; Kim, H.Y.; Park, J.; Kim, J. Compact spatial pyramid pooling deep convolutional neural network based hand gestures decoder. Appl. Sci. 2020, 10, 7898.

Kim, S.Y.; Han, H.G.; Kim, J.W.; Lee, S.; Kim, T.W. A Hand Gesture Recognition Sensor Using Reflected Impulses. IEEE Sens. J. 2017, 17, 2975–2976

Rosenberger, P.; Cosgun, A.; Newbury, R.; Kwan, J.; Ortenzi, V.; Corke, P.; Grafinger, M. Object-Independent Human-to-Robot Handovers Using Real Time Robotic Vision. IEEE Robot. Autom. Lett. 2021, 6, 17–23.

Woo, S.-W.; Kang, M.-K.; Hong, K.-S. Classification of Finger Tapping Tasks using Convolutional Neural Network Based on Augmented Data with Deep Convolutional Generative Adversarial Network. In Proceedings of the 2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob), New York, NY, USA, 29 November–1 December 2020; Volume 9224386, pp. 328–333.

Chawla, B.; Tyagi, S.; Jain, R.; Talegaonkar, A.; Srivastava, S. Finger Vein Recognition Using Deep Learning. Adv. Intell. Syst.Comput. 2021, 1164, 69–78.

LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. Neural Comput. 1989, 1, 541–551,

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, 18–23 June 2018; pp. 4510–4520.

Howard, A.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Hartwig, A. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv 2017, arXiv:1704.04861.

Convolutional Pose Machines Tensorflow. [Online]. Available on: https://github.com/timctho/convolutional-pose-machines-tensorflow. Accessed: June 2022.

Wei, S.; Ramakrishna, V.; Kanade, T.; Sheikh, Y. Convolutional Pose Machines. arXiv 2016, arXiv:1602.00134.

Hamidieh, K. (2018). A data-driven statistical model for predicting the critical temperature of a superconductor. Com-putational Materials Science,

Noe, M., & Steurer, M. (2007). High-temperature superconductor fault current limiters: concepts, applications, and de-velopment status. Superconductor Science and Technology, 20(3), R15–R29.

Krinner, S., Storz, S., Kurpiers, P., Magnard, P., Heinsoo, J., Keller, R., Wallraff, A. (2019). Engineering cryogenic setups for 100-qubit scale superconducting circuit systems. EPJ Quantum Technology, 6(1).

Si, Q., Yu, R., Abrahams, E. (2016). High-temperature superconductivity in iron pnictides and chalcogenides. Nature Reviews Materials, 1(4).

Flores-Livas, J. A., Sanna, A., Gross, E. K. U. (2016). High temperature superconductivity in sulfur and selenium hy-drides at high pressure. The European Physical Journal B, 89(3).

Sleight, A. W., Gillson, J. L., Bierstedt, P. E. (1993). High-temperature superconductivity in the BaPb1−xBixO3 system. Solid State

Communications, 88(11-12), 841–842.

Abdulkadir, K., Kemal, A., (2019), Predicting the Critical Temperature of Superconductors with XGBoost. Conference paper, ICAIAME 2019, 74-76

Shaobo Li, Yabo Dan, Xiang Li, (2019). Critical Temperature Prediction of Superconductors Based on Atomic Vectors and Deep Learning. Symmetry 2019, 12, 262

Center for Machine Learning and Intelligent Systems, Bren School of Information and Computer Science, University of California, Irvine, 2020. https://archive.ics.uci.edu/ml/datasets/Superconductivty+Data

Periodic Table of Elements. [Online]. Available on: https://gist.github.com/GoodmanSciences/c2dd862cd38f21b0ad36b8f96b4bf1ee. Accessed: June 2022.

Understanding LSTM Networks. [Online]. Available on: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Accessed: June 2022.

Alam, K.M.R., Siddique, N. & Adeli, H. (2019). A dynamic ensemble learning algorithm for neural networks. Neural Comput & Applic .

Z. P. Lu, Y. Li, S. C. Ng, Journal of Non-Crystalline Solids, Reduced glass transition temperature and glass forming ability of bulk glass forming alloys, 270(1-3), 103–114, (2000).

H. Wu, T. Mayeshiba, D. Morgan, Scientific Data, High-throughput ab-initio dilute solute diffusion database, 3, (2016)

L. Ward, A. Agrawal, A. Choudhary, C. Wolverton, Npj Computational Materials, A General-Purpose Machine Learning Framework for Predicting Properties of Inorganic Materials, 1–7, (2015).

W. Wayne, Applied Nonparametric Statistics (2nd ed.), Kolmogorov–Smirnov one-sample test, pp. 319–330, (1990).

K. Ramasubramanian, A. Singh, Deep Learning Using Keras and TensorFlow, (2018)

M. Shanker, M. Hu,. Omega, Effect of data standardization on neural network training, 24(4), 385–397. (1996)

N. Vo, J. Hays, 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Generalization in Metric Learning: Should the Embedding Layer Be Embedding Layer?, (2019).

G. Yarin, H. Jiri, A. Kendall, Concrete Dropout, (2017).

D. Hawkins, Journal of Chemical Information and Computer Sciences, The

Problem of Overfitting. 44(1), 1–12, (2004).

Zhu, H., Luo, W., Ciesielski, P. N., Fang, Z., Zhu, J. Y., Henriksson, G., Hu, L. (2016). Wood-Derived Materials for Green Electronics, Biological Devices, and Energy Applications. Chemical Reviews, 116(16), 9305–9374.

Mosier, N. (2005). Features of promising technologies for pretreatment of lignocellulosic biomass. Bioresource Technology, 96(6), 673–686.

Abdul Khalil, H. P. S., Davoudpour, Y., Islam, M. N., Mustapha, A., Sudesh, K., Dungani, R., & Jawaid, M. (2014). Production and modification of nanofibrillated cellulose using various mechanical processes: A review. Carbohydrate Polymers, 99, 649–665.

Lavoine, N., Desloges, I., Dufresne, A., & Bras, J. (2012). Microfibrillated cellulose – Its barrier properties and applications in cellulosic materials: A review. Carbohydrate Polymers, 90(2), 735–764.

Klemm, D., Heublein, B., Fink, H.-P., & Bohn, A. (2005). Cellulose: Fascinating Biopolymer and Sustainable Raw Material. Angewandte Chemie International Edition, 44(22), 3358–3393.

Himanen, L., Geurts, A., Foster, A. S., & Rinke, P. (2019). Data-Driven Materials Science: Status, Chal-lenges, and Perspectives. Advanced Science, 1900808.

Dresselhaus, M. S., Crabtree, G. W., & Buchanan, M. V. (2005). Addressing Grand Energy Challenges through Advanced Materials. MRS Bulletin, 30(07)

Stach, E., DeCost, B., Kusne, A. G., Hattrick-Simpers, J., Brown, K. A., Reyes, K. G., … Maruyama, B. (2021). Autonomous experimentation systems for materials development: A community perspective. Matter, 4(9), 2702–2726

Pintsuk, G., Diegele, E., Dudarev, S. L., Gorley, M., Henry, J., Reiser, J., & Rieth, M. (2019). European materials development: Results and perspect ive. Fu-sion Engineering and Design

Kipouros, G. J., Caley, W. F., & Bishop, D. P. (2006). On the advantages of using powder metallurgy in new light metal alloy design. Metallurgical and Materials Transactions, 37(12), 3429–3436

Hardian, R., Liang, Z., Zhang, X., & Szekely, G. (2020). Artificial intelligence: the silver bullet for sustainable materials development. Green Chemistry

Ball, P. (2019). Using artificial intelligence to accel-erate materials development. MRS Bulletin, 44(05), 335–344.

Sha, W., Guo, Y., Yuan, Q., Tang, S., Zhang, X., Lu, S., … Cheng, S. (2020). Artificial Intelligence to Power the Future of Materials Science and Engineering. Ad-vanced Intelligent Systems, 1900143

Wei, J., Chu, X., Sun, X., Xu, K., Deng, H., Chen, J., … Lei, M. (2019). Machine learning in materials science. InfoMat, 1(3), 338–358

Huang, J. S., Liew, J. X., Ademiloye, A. S., & Liew, K. M. (2020). Artificial Intelligence in Materials Model-ing and Design. Archives of Computational Methods in Engineering, 28(5), 3399–3413

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative Adversarial Networks: An Overview. IEEE Signal Processing Magazine, 35(1), 53–65

Wang, K., Gou, C., Duan, Y., Lin, Y., Zheng, X., & Wang, F.-Y. (2017). Generative adversarial networks: introduction and outlook. IEEE/CAA Journal of Au-tomatica Sinica, 4(4), 588–598

Mechanical properties of low alloy steels. [Online]. Available on: https://www.kaggle.com/rohannemade/mechanical-properties-of-low-alloy-steels. Accessed: June 2022.

Li L., Jamieson K., DeSalvo G., Rostamizadeh A., Talwalkar A. (2018), Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization, Journal of Machine Learning Research, 18, 1-52, https://doi.org/10.48550/arXiv.1603.06560