1. **Introduction to Carbon Fibre reinforced Plastic Composites**

[Carbon Fibre](https://www.thoughtco.com/what-is-carbon-fiber-820397) Reinforced Polymer Composites (CFRP) are lightweight, strong materials used in the manufacturing of numerous products used in our daily life. It is a term used to describe a fibre-reinforced [composite material](https://www.thoughtco.com/what-is-a-composite-820406) that uses carbon fibre as the primary structural component. It should be noted that the "P" in CFRP can also stand for "plastic" instead of "polymer." In general, CFRP composites use thermosetting resins such as epoxy, [polyester, or vinyl ester](https://www.thoughtco.com/vinyl-ester-vs-polyester-resins-820376). Although [thermoplastic resins](https://www.thoughtco.com/thermoplastic-vs-thermoset-resins-820405) are used in CFRP Composites, "Carbon Fiber Reinforced Thermoplastic Composites" often go by their own acronym, CFRTP composites.

 When working with composites or within the composites industry, it is important to understand the terms and acronyms. More importantly, it is necessary to understand the [properties of FRP composites](https://www.thoughtco.com/properties-of-frp-composites-820515) and capabilities of the various reinforcements such as carbon fiber.The use of carbon fibers in plastic materials has a long history.  As early as 1879, Thomas Edison was experimenting with carbon fibers made from cotton threads and bamboo slivers.  In fact, the first incandescent light bulb heated by electricity contained carbon fibers.

In the 1960’s, Dr. Akio Shindo at the Agency of Industrial Science and Technology in Japan developed a carbon fiber based on polyacrylonitrile (PAN).  The resulting fiber contained 55% carbon. The PAN-based conversion process quickly became the primary method for producing [carbon fiber](https://www.craftechind.com/beginners-guide-fiber-reinforced-plastics-frps/).  Ninety percent of carbon fibers today are made from [polyacrylonitrile](https://en.wikipedia.org/wiki/Polyacrylonitrile) (C3H3N)n  or PAN a synthetic, semi-crystalline organic polymer resin.The remaining 10% are made from rayon or petroleum pitch. Fibers made from PAN are extremely strong and light. These fibers are bound by thermoset or thermoplastic polymers such as polyester, vinyl ester or nylon to make carbon fiber reinforced plastic, or carbon FRP.

##### **Adding Carbon Fiber to a Polymer Has Many Benefits**

Tensile strength and flexural modulus are increased as is the [heat](https://www.craftechind.com/dont-sweat-4-high-temp-plastics-can-take-heat/) deflection temperature or HDT.  Additionally, adding carbon fiber reinforcement diminishes shrinkage and warping.  Each carbon fiber is a long thin strand made up of thousands of [carbon filaments](https://www.craftechind.com/blog/a-closer-look-at-glass-fibers-in-reinforced-plastic). A single fiber is about 5-10 μm in diameter and composed mostly of carbon.  Microscopic crystals in the carbon bond together in a structure that is more or less aligned parallel to the long axis of the fiber.  It is this alignment of crystals that make the fibers so strong.

##### **Classified by Tensile Modulus**

Carbon fibers are classified by the tensile modulus**\*** of the fiber.  The tensile modulus may range from 34.8 million psi to 72.5-145.0 million psi.  Steel has a tensile modulus of 29 million psi thus the strongest carbon fiber is five times stronger than steel. “Low” modulus fibers have a tensile modulus below 34.8 million psi (240 million kPa). Fibers are also classified in ascending order of tensile modulus as “standard modulus,” “intermediate modulus,” “high modulus,” and “ultrahigh modulus.” Carbon fibers with a classification of ultrahigh modulus have a tensile modulus of 72.5-145.0 million psi (500 million-1.0 billion kPa).

##### **Spinning, Stabilizing, Carbonizing, Surface Treatment and Sizing**

The manufacturing process for carbon fiber is partly chemical and partly mechanical.

* **Spinning:** The PAN is spun using one of a few spinning processes.  This step is important because it forms the internal atomic structure of the fiber.  The fibers are then washed and stretched to the required diameter.  The stretching also helps align the molecules to aid in the formation of the carbon crystals created by carbonization.
* **Stabilizing:**In this step the fibers are treated with chemicals to change their linear bonding to a thermally stable ladder bonding structure.  The filaments are then heated in air so they pick up oxygen molecules and change their atomic bonding pattern.
* **Carbonizing:** The fibers are then exposed to very high heat without oxygen present so the fiber cannot burn.  The atoms in the fiber vibrate violently expelling most of the non-carbon atoms in the precursor.
* **Surface Treatment:**  After carbonizing, the surface of the fibers does not bond well with the materials used in making composite materials.  In this step, the surface of the fibers are slightly oxidized by immersion in various gases or liquids.
* **Sizing:**In this process, the fibers are coated to protect them from damage during winding or weaving.

## **Properties of CFRP Composites**

Composite materials, reinforced with carbon fiber, are different than other FRP composites using traditional materials such as fiberglass or [aramid fiber](https://www.thoughtco.com/aramid-fibers-definition-820379). The properties of CFRP composites that are advantageous include:

**Light Weight:**

A traditional [fiberglass reinforced composite](https://www.thoughtco.com/what-is-fiberglass-or-glass-fiber-820469) using continuous glass fiber with a fiber of 70% glass (weight of glass / total weight), will commonly have a density of .065 pounds per cubic inch.

Meanwhile, a CFRP composite, with the same 70% fiber weight, might typically have a density of .055 pounds per cubic inch.

**Increased Strength:**

Not only are carbon fiber composites lighter weight, but CFRP composites are much stronger and stiffer per unit of weight. This is true when comparing carbon fiber composites to glass fiber, but even more so when compared to metals. For example, a decent rule of thumb when comparing steel to CFRP composites is that a carbon fiber structure of equal strength ​will often weigh 1/5th that of steel. You can imagine why automotive companies are investigating using carbon fiber instead of steel.

When comparing CFRP composites to aluminum, one of the lightest metals used, a standard assumption is that an aluminum structure of equal strength would likely weigh 1.5 times that of the carbon fiber structure. Of course, there are many variables that could change this comparison. The grade and quality of materials can be different, and with composites, the [manufacturing process](https://www.thoughtco.com/how-is-carbon-fiber-made-820391), fiber architecture, and the quality need to be taken into account.

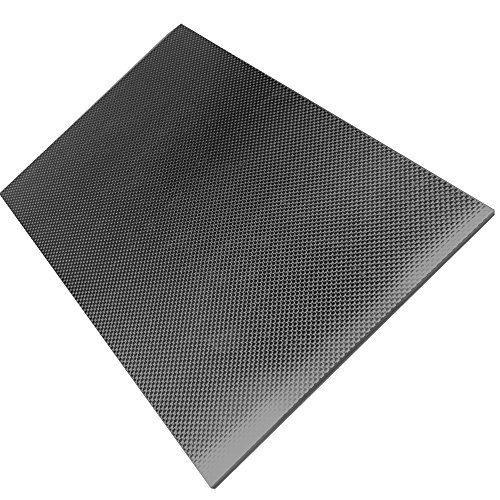
**Cost:**

Although amazing material, there is a reason why carbon fiber is not used in every single application. At the moment, CFRP composites are cost-prohibitive in many instances. Depending on the current market conditions (supply and demand), the type of carbon fiber (aerospace vs. commercial grade), and the fiber tow size, the price of carbon fiber can vary dramatically. Raw carbon fiber on a price-per-pound basis can be anywhere between 5-times to 25-times more expensive than fiberglass. This disparity is even greater when comparing steel to CFRP composites.

**Conductivity:**

This can be both an advantage to carbon fiber composites, or a disadvantage depending on the application. Carbon fiber is extremely conductive, while glass fiber is insulative. Many [applications use glass fiber](https://www.thoughtco.com/uses-of-fiberglass-820412), and cannot use carbon fiber or metal, strictly because of the conductivity. For example, in the utility industry, many products are required to use glass fibers. It is also one of the reasons why ladders use glass fiber as the ladder rails. If a fiberglass ladder were to come in contact with a power line, the chances of electrocution are much lower. This would not be the case with a CFRP ladder.

Although the cost of CFRP composites still remains high, new technological advancements in manufacturing are continuing to allow for more cost-effective products. Hopefully, in our lifetime, we will be able to see cost-effective carbon fiber used in a wide range of consumer, industrial, and automotive applications.

**(Schematic of CFRP)**

1. **Abstract:**

Delamination within aerospace composites are of particular concern, presenting within composite laminate structures without visible surface indications. Transmission based thermography techniques using contact temperature sensors and surface mounted heat sources are able to detect reductions in thermal conductivity and in turn impact damage and large dis bonds can be detected. However, delamination’s between Carbon Fibre Reinforced Polymer (CFRP) plies are not immediately discoverable using the technique. The use of transient thermal conduction profiles induced from zonal heating of a CFRP laminate to ascertain inter-laminate differences has been demonstrated and the project builds on this method further by investigating the impact of inter laminate inclusions, in the form of delamination’s, to the transient thermal conduction profile of multi-ply bi-axial CFRP laminates.

Results demonstrate that as the distance between centre of the heat source and delamination increase, whilst maintaining the delamination within the heated area, the resultant transient thermal conduction profile is measurably different to that of a homogeneous region at the same distance. The method utilises a supervised Support Vector Classification (SVC) algorithm and Artificial Neural Network with TensorFlow to detect delamination using temperature data from either the edge of the defect or the centre during a 140 s ramped heating period to 80 o C. An F1 score in the classification of delamination or no delamination at an overall accuracy of over 99% in both training and with test data separate from the training process has been achieved using data points effected by transient thermal conduction due to structural dissipation at 56.25 mm.

**3. Objective:**

1. Understanding basic ply and composition of typical Carbon Fibre reinforced plastic composites.

2. Understanding how delamination relates to defect in terms of failure analysis

3.Understanding Transmission based thermography techniques, thermal conduction profiles and how this experiment outputs are becoming features of the dataset which will be fed into Artificial Intelligence Unit to make prediction (defect).

4.Develop AI model using Support Vector Machining Algorithm

5.Develop AI model using Deep Neural Network using TensorFlow

6.Compare and Contrast both models’ performance using Accuracy score.

7.Save the best model for further Exploration on inference.

1. **Literature Review:**

The following table describes the research papers and the corresponding inference.

|  |  |  |  |
| --- | --- | --- | --- |
| Title | Author | volume, Issue, Year, Journal | Inference |
| Composite laminate delamination detection using Transient Thermal Conduction  Profiles and Machine Learning Based Data Analysis. | David I. Gillespie et.al | [www.mdpi.com/journal/sensors](http://www.mdpi.com/journal/sensors)  2020, 20, 7227 | Through the paper clearly understood how CFRP is composed of and importance of transient thermal conduction profiles towards dataset preparation for AI modelling. |
| Machine learning approach for delamination detection with feature missing and noise polluted vibration characteristics | Yushu Li et al | [Volume 287](https://www.sciencedirect.com/journal/composite-structures/vol/287/suppl/C),  1 May 2022, 115335 [Composite Structures](https://www.sciencedirect.com/journal/composite-structures) | Understanding how machine learning models are supporting defect detection through various fundamental statistical algorithms. |
| A comparison of machine learning algorithms for assessment of delamination in fiber-reinforced polymer composite beams | Yshuo Wang et al | [Volume: 20 issue: 4,](https://journals.sagepub.com/toc/shm/20/4)page(s): 1997-2012  Structural Health Monitoring | Understanding how a delamination detection methodology is proposed based on the changes in multiple modes of frequencies to assess the interface, location, and size of delamination in fiber-reinforced polymer composites. Three types of machine learning algorithms including back propagation neural network, extreme learning machine, and support vector machine algorithm were adopted as inverse algorithms for assessment of the delamination parameters, with a special focus on the interface prediction |
| Artificial intelligence techniques for fault assessment in laminated composite structure: a review | Sidharth Patro et al | E3S Web of Conferences 309,  01083 (2021) | The present paper aims to bring out a concise review on various methodologies employed for damage/fault detection in composite materials with a special emphasis on supervised and unsupervised machine learning techniques. The major observations are outlined with an objective to put forward a broad perspective of the state of art related to laminated composite structural heath monitoring. |
| Delamination prediction in composite panels using unsupervised-feature learning methods with wavelet-enhanced guided wave representations | [MahindraRautela](https://www.sciencedirect.com/science/article/abs/pii/S026382232200366X?via%3Dihub" \l "!)  Et al. | [Volume 291](https://www.sciencedirect.com/journal/composite-structures/vol/291/suppl/C), 1 July 2022, 115579 [Composite Structures](https://www.sciencedirect.com/journal/composite-structures) | In this paper, researcher proposed two different unsupervised-feature learning approaches where the algorithms are trained only on the baseline scenarios to learn the distribution of [baseline signals](https://www.sciencedirect.com/topics/engineering/baseline-signal). The trained unsupervised feature learner is used for delamination prediction with an [anomaly detection](https://www.sciencedirect.com/topics/engineering/anomaly-detection) philosophy. In the first approach, and combined [dimensionality reduction techniques](https://www.sciencedirect.com/topics/engineering/dimensionality-reduction-technique) (principal component analysis and independent component analysis) with a one-class [support vector machine](https://www.sciencedirect.com/topics/engineering/support-vector-machine). |

1. **Methodology:**

Study on Carbon Fibre reinforced plastic fabrication with fault layer

Study on Experimentation through Transmission based thermography techniques

Study on obtaining transient thermal conduction profiles

Create Dataset for Data Modelling from transient thermal conduction profiles

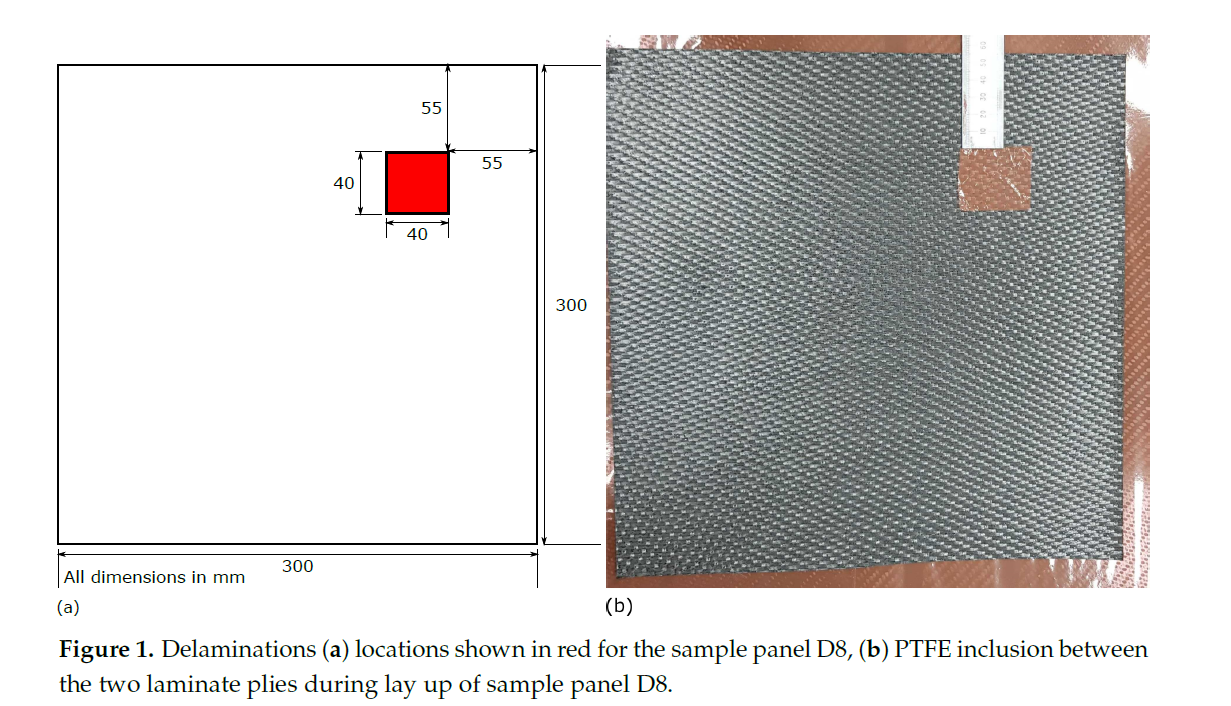
Apply Feature Engineering on Dataset and Finalize Dataset

Compare and contrast AI models and choose the best

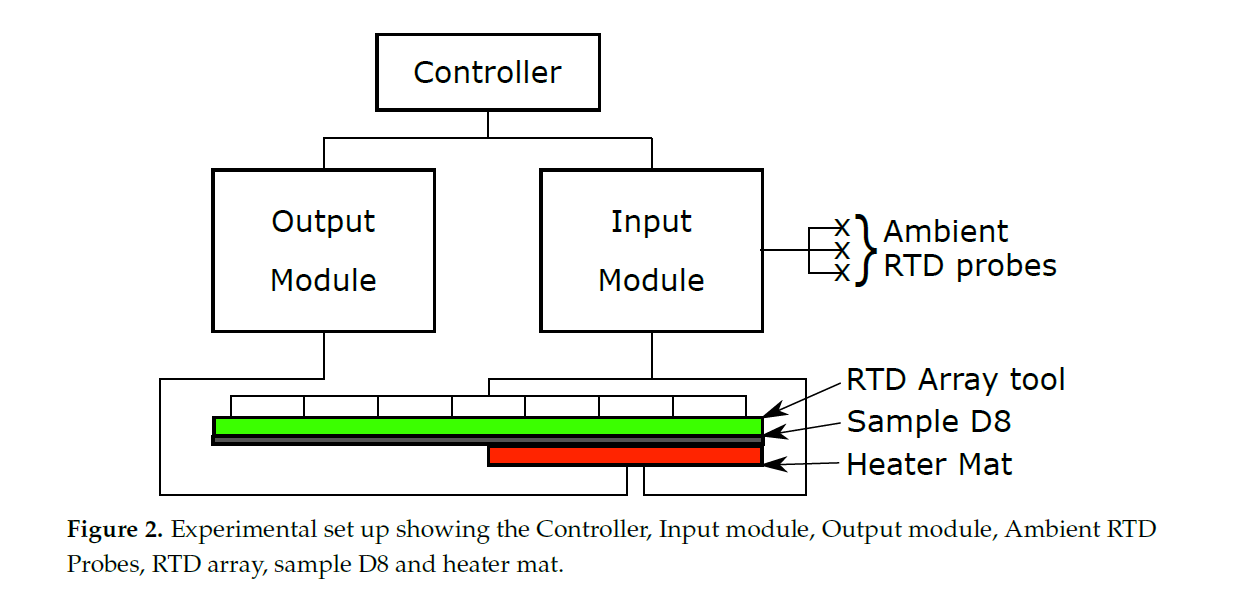
Implement and Develop Artificial Intelligence Models using Machine Learning Algorithm and Deep Learning Neutral Network

1. **Study on Carbon Fibre reinforced plastic fabrication with fault layer and Experimentation through Transmission based thermography techniques:**

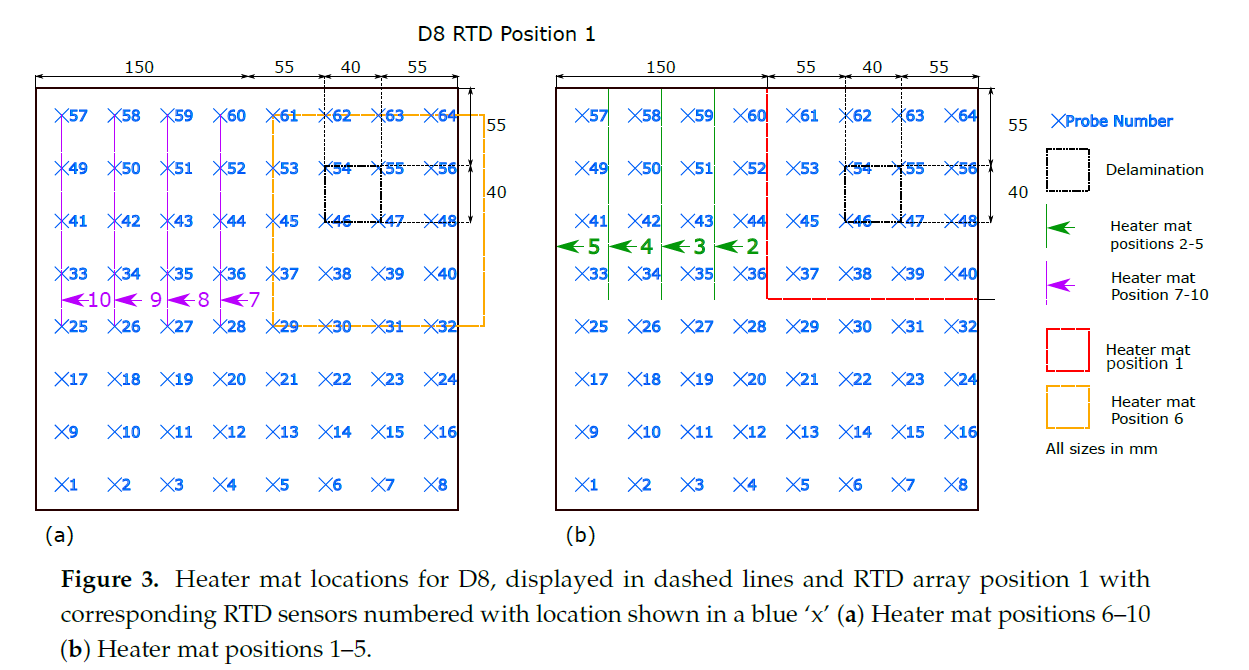
A CFRP sample laminate of 300 x 300 mm in size was created from bi-axial CFRP 5 harness ply with density 1589 kg /m, two plies were laid up with orientation 0/90/0/90, 0/90/0/90 to produce non-symmetrical and balanced laminate samples of 0.7366 mm in thickness. Creation of a delamination within the sample was achieved through the insertion of two Polytetrafluoroethylene (PTFE) squares 40 x 40 mm, of 0.01 mm thickness between the ply boundary of the two plies of the laminate sample at the top and right edges 55 mm from the top and right-hand edges. The Sample was cured within Collins Aerospace’s Nitrogen pressurised Autoclave at 180 Degree C for 2 h at 35 psi. The cured samples were inspected via the tap testing method by an experienced aerospace engineer in order to ensure that a delamination had occurred. As the sample was the 8th defect variation type within a larger body of work, and in order to maintain the naming convention with the associated published data set it was assigned the label “D8” so as to allow the identification of the corresponding open data sets.



An RTD array of RS Pro Thin Film Pt100 Platinum Resistance Temperature Detector (RTD) contact temperature sensors arranged in an 8 x 8 array with probes spaced evenly in the x and y axis by 37.5 mm, was positioned in one of two positions on sample D8, with a 150 x 150 mm heater mat positioned on the opposite surface to that of the RTD array Figure 2. The RTD temperatures were recorded by a 4-channel analogue input with the heater mat temperature powered through an output module, the temperature of which being controlled via an RTD applied to the heater mat. Ambient temperature was also recorded through 3 further RTDs attached to the analogue input module which were positioned away from the rest of the experimental set up.

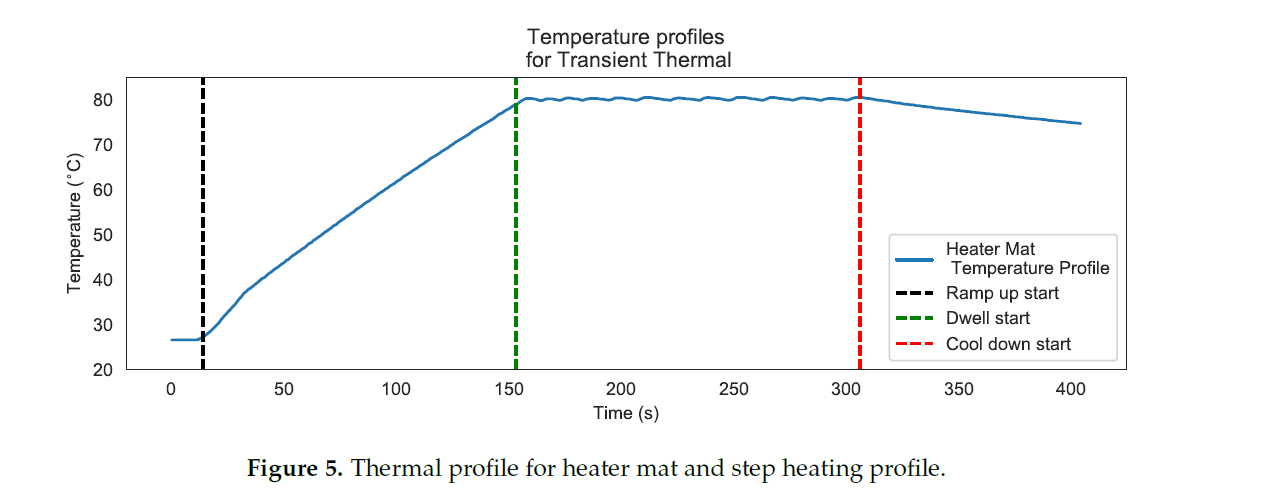


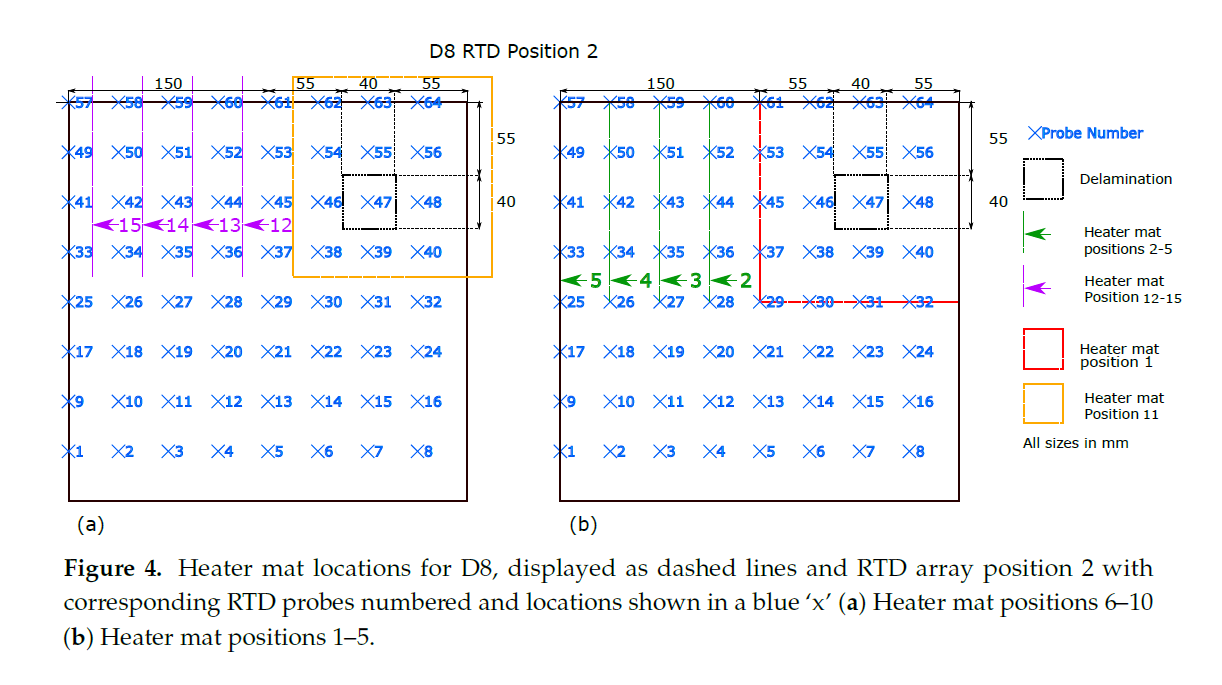
RTD position 1 (Figure 3) represents the RTD array positioned centrally on the sample with an edge distance around the RTD array of 18.75 mm. The delamination is directly under 4 RTD probes, 46, 47, 54 and 55 at this RTD array position. As shown in Figure 3b a 150 x 150 mm heater mat is located in the top right quadrant of sample D8 and is applied to the opposite surface of the sample to that of the RTD array. The initial position (Heater mat position 1) matches the distance of the centre of the heater mat to the centre of the delamination. A further four heater mat positions are created for testing, shown in green, with a relocation of the heater mat in 37.5 mm increments towards the left-hand edge of sample D8, yielding five unique data capture conditions. Using the same spacing between the five heater mat locations but altering the initial position by 18.75 mm in the y-axis and +18.75 mm in x-axis (Figure 3b) provides another five data capture conditions.



The heater mat positions 1–5 at RTD position one (Figure 3b) were replicated at RTD Position2 shown in in Figure 4b. However, the RTD array itself was relocated 􀀀18.75 mm in the x-axis and +18.75 mm in the y-axis. The relocation of the RTD array provided distances from the RTD sensors corresponding to those shown in Figure 3a heater mat positions 6–10, providing directly comparable readings. Replicating the heater mat positions 6–10 at RTD array position 2 in relation to the sample D8, produce distances from the RTD sensors corresponding to those at RTD array position 1 heater mat positions 6–10. In order to create heater mat locations in relation to sample D8 that are directly comparable to RTD array position 1 heater mat positions 1–5, a further set of heater mat positions were required, achieved by creating heater mat positions 11–15 (Figure 4a) which were offset from heater mat positions 1–5 by 􀀀18.75 mm in the x-axis and +18.75 mm in the y-axis. As shown, 4 directly comparable RTD distance from heat source positions (1–4 and 12–15) result, however positions 11 and 5 have no directly comparable heater mat positions in regards to RTDs. Thus, the maximum quantity of directly comparable heater mat positions are established whilst maintaining the minimum starting edge distance of the delamination to the heat source of 37.5 mm (as seen in heater mat position 6 and 11, Figures 3a and 4a respectively). The trade-off permits the maximum directly comparable RTD to heater mat positions, whilst maintaining the transient thermal conduction profiles in the x-axis of CFRP impacted by the delamination.

Temperature against time profiles were captured and labelled according to RTD locations relative to the heater mat location and sensor positions over a delamination. In order to minimise the risk of expansion to the manufactured delamination, or the creation of new defects due to thermal cyclic loading, a low energy, low temperature (maximum 80 Degree C) heating profile was applied. The low temperature stimulus was segmented into three distinct phases: Ramp up; Dwell; and Cool down (Figure 5) over periods of 140 s, Ramp up; 160 s, Dwell; and 90 s Cool down respectively. Sample D8 was heated using the stepped heating profile at each of the heater mat locations and RTD array positions.

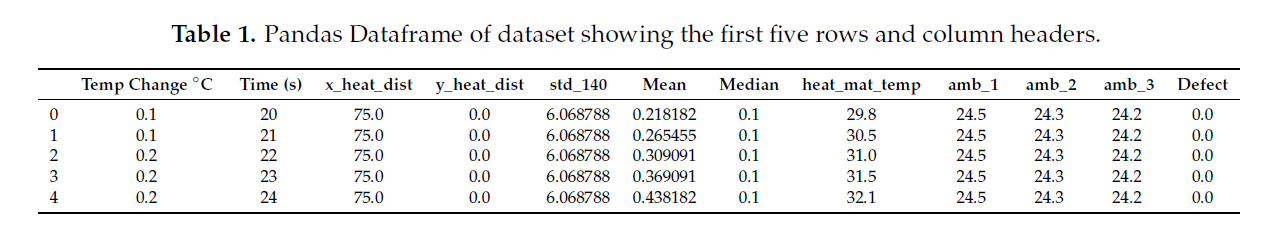




1. **Create Dataset for Data Modelling from transient thermal conduction profiles:**

For heater mat positions displayed in Figures 3 and 4, sample D8 was subjected to 10 Ramp, Dwell and Cool down for RTD positions 1 and 2 repeated 10 times for each unique heater mat/RTD position combination, resulting in 12,800 temperature against time profiles. The resultant temperature. against time profiles were plotted together to surface obvious outliers. The first stage in collating useful data was to remove any temperature profiles which displayed a reduction in temperature during the ramp and dwell stages, an indication of a faulty RTD sensor. The profiles were then examined visually in batches of 100 to identify profiles displaying excessive levels of noise that would be suggest that an RTD had failed or a connection issue between the probe and recording equipment. The remaining profiles had three annotations assigned to each: ‘x\_heat\_dist’, the distance that the probe is currently from the centre of the heat mat in the x-axis; ‘y\_heat\_dist’, the distance that the probe is currently from the centre of the heat mat in the y-axis; ‘defect’, if the probe is currently placed above a homogeneous region ‘0’, the edge of a delamination ‘1’, at the centre of a delamination ‘2’, or finally if the probe is above a homogeneous region where there is however a defect between the probe and the centre of the heater mat ‘3’. The profiles where segmented further into single points of ‘Temp Change Degree C’ against ‘Time (s)’ yielding an appropriately cleansed data set of 3,833,960 rows. Three further features where added to these single points; ambient temperature readings in Degree C taken at the same instance as the corresponding data entry (‘amb\_1’, ‘amb\_2’ and ‘amb\_3’). The use of an RTD to control the temperature within the heater mat through a feedback loop also provides another feature in the form of the ‘heater mat temperature’ in \_C at the corresponding data entry time interval for each sample run (‘heat\_mat\_temp’). Given that all temperature readings are known for each sample at every time instance allows for the annotation of two additional features; the mean and median values, as these can be easily calculated for each time instance (‘mean’ and ‘median’ respectively).

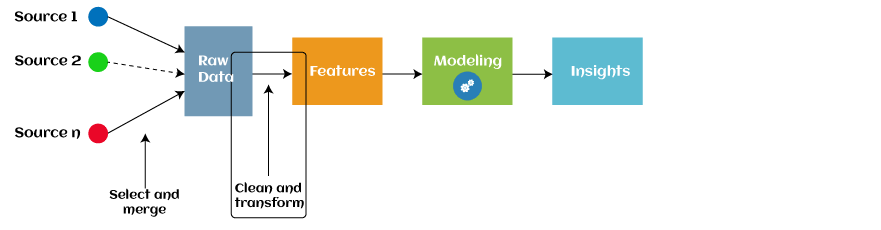
The impact of a delamination on the transient thermal conduction recorded by an RTD through temperature profiles is only captured by RTDs which are in line with the defect on the x-axis of the technique. The instance of RTD position 1 (Figure 3) relates to RTDs 41–56, and for RTD position 2 (Figure 4) RTDs 41–48. As the most apparent differences between defect types occurs within the 20–160 s time period, the data set can be reduced to only consider values from this period. Inspection of the captured temperature against time profile for each RTD probe within the 20–160 s time period also allows calculation for the standard deviation, the final feature to the data set (‘std\_140’). However, restricting the data set within this range reduces the overall data set for training and validation of the machine learning algorithm to 259,299. Table 1 displays the initial five rows of the data set under the feature headings.



1. **Apply Feature Engineering on Dataset and Finalize Dataset:**

Generally, all machine learning algorithms take input data to generate the output. The input data remains in a tabular form consisting of rows (instances or observations) and columns (variable or attributes), and these attributes are often known as **features**. For example, an image is an instance in computer vision, but a line in the image could be the feature. Similarly, in NLP, a document can be an observation, and the word count could be the feature. So, we can say **a feature is an attribute that impacts a problem or is useful for the problem**.

**Feature engineering is the pre-processing step of machine learning, which extracts features from raw data**. It helps to represent an underlying problem to predictive models in a better way, which as a result, improve the accuracy of the model for unseen data. The predictive model contains predictor variables and an outcome variable, and while the feature engineering process selects the most useful predictor variables for the model.



Since 2016, automated feature engineering is also used in different machine learning software that helps in automatically extracting features from raw data. Feature engineering in ML contains mainly four processes: **Feature Creation, Transformations, Feature Extraction, and Feature Selection.**

These processes are described as below:

1. Feature Creation: Feature creation is finding the most useful variables to be used in a predictive model. The process is subjective, and it requires human creativity and intervention. The new features are created by mixing existing features using addition, subtraction, and ration, and these new features have great flexibility.
2. Transformations: The transformation step of feature engineering involves adjusting the predictor variable to improve the accuracy and performance of the model. For example, it ensures that the model is flexible to take input of the variety of data; it ensures that all the variables are on the same scale, making the model easier to understand. It improves the model's accuracy and ensures that all the features are within the acceptable range to avoid any computational error.
3. Feature Extraction: Feature extraction is an automated feature engineering process that generates new variables by extracting them from the raw data. The main aim of this step is to reduce the volume of data so that it can be easily used and managed for data modelling. Feature extraction methods include cluster analysis, text analytics, edge detection algorithms, and principal components analysis (PCA).
4. Feature Selection: While developing the machine learning model, only a few variables in the dataset are useful for building the model, and the rest features are either redundant or irrelevant. If we input the dataset with all these redundant and irrelevant features, it may negatively impact and reduce the overall performance and accuracy of the model. Hence it is very important to identify and select the most appropriate features from the data and remove the irrelevant or less important features, which is done with the help of feature selection in machine learning. "Feature selection is a way of selecting the subset of the most relevant features from the original features set by removing the redundant, irrelevant, or noisy features."

Below are some benefits of using feature selection in machine learning:

* It helps in avoiding the curse of dimensionality.
* It helps in the simplification of the model so that the researchers can easily interpret it.
* It reduces the training time.
* It reduces overfitting hence enhancing the generalization.

## Need for Feature Engineering in Machine Learning:

In machine learning, the performance of the model depends on data pre-processing and data handling. But if we create a model without pre-processing or data handling, then it may not give good accuracy. Whereas, if we apply feature engineering on the same model, then the accuracy of the model is enhanced. Hence, feature engineering in machine learning improves the model's performance. Below are some points that explain the need for feature engineering:

* **Better features mean flexibility.**  
  In machine learning, we always try to choose the optimal model to get good results. However, sometimes after choosing the wrong model, still, we can get better predictions, and this is because of better features. The flexibility in features will enable you to select the less complex models. Because less complex models are faster to run, easier to understand and maintain, which is always desirable.
* **Better features mean simpler models.**  
  If we input the well-engineered features to our model, then even after selecting the wrong parameters (Not much optimal), we can have good outcomes. After feature engineering, it is not necessary to do hard for picking the right model with the most optimized parameters. If we have good features, we can better represent the complete data and use it to best characterize the given problem.
* **Better features mean better results.**  
  As already discussed, in machine learning, as data we will provide will get the same output. So, to obtain better results, we must need to use better features.

## Steps in Feature Engineering:

The steps of feature engineering may vary as per different data scientists and ML engineers. However, there are some common steps that are involved in most machine learning algorithms, and these steps are as follows:

* **Data Preparation:** The first step is data preparation. In this step, raw data acquired from different resources are prepared to make it in a suitable format so that it can be used in the ML model. The data preparation may contain cleaning of data, delivery, data augmentation, fusion, ingestion, or loading.
* **Exploratory Analysis:** Exploratory analysis or Exploratory data analysis (EDA) is an important step of features engineering, which is mainly used by data scientists. This step involves analysis, investing data set, and summarization of the main characteristics of data. Different data visualization techniques are used to better understand the manipulation of data sources, to find the most appropriate statistical technique for data analysis, and to select the best features for the data.
* **Benchmark**: Benchmarking is a process of setting a standard baseline for accuracy to compare all the variables from this baseline. The benchmarking process is used to improve the predictability of the model and reduce the error rate.

## Feature Engineering Techniques:

Some of the popular feature engineering techniques include:

### 1. Imputation

Feature engineering deals with inappropriate data, missing values, human interruption, general errors, insufficient data sources, etc. Missing values within the dataset highly affect the performance of the algorithm, and to deal with them "Imputation" technique is used. **Imputation is responsible for handling irregularities within the dataset.**

For example, removing the missing values from the complete row or complete column by a huge percentage of missing values. But at the same time, to maintain the data size, it is required to impute the missing data, which can be done as:

* For numerical data imputation, a default value can be imputed in a column, and missing values can be filled with means or medians of the columns.
* For categorical data imputation, missing values can be interchanged with the maximum occurred value in a column.

### 2. Handling Outliers

Outliers are the deviated values or data points that are observed too away from other data points in such a way that they badly affect the performance of the model. Outliers can be handled with this feature engineering technique. This technique first identifies the outliers and then remove them out.

**Standard deviation** can be used to identify the outliers. For example, each value within a space has a definite to an average distance, but if a value is greater distant than a certain value, it can be considered as an outlier. **Z-score** can also be used to detect outliers.

### 3. Log transform

Logarithm transformation or log transform is one of the commonly used mathematical techniques in machine learning. Log transform helps in handling the skewed data, and it makes the distribution more approximate to normal after transformation. It also reduces the effects of outliers on the data, as because of the normalization of magnitude differences, a model becomes much robust.

### 4. Binning

In machine learning, overfitting is one of the main issues that degrade the performance of the model and which occurs due to a greater number of parameters and noisy data. However, one of the popular techniques of feature engineering, "binning", can be used to normalize the noisy data. This process involves segmenting different features into bins.

### 5. Feature Split

As the name suggests, feature split is the process of splitting features intimately into two or more parts and performing to make new features. **This technique helps the algorithms to better understand and learn the patterns in the dataset.**

The feature splitting process enables the new features to be clustered and binned, which results in extracting useful information and improving the performance of the data models.

### 6. One hot encoding

One hot encoding is the popular encoding technique in machine learning. It is a technique that converts the categorical data in a form so that they can be easily understood by machine learning algorithms and hence can make a good prediction. It enables group the of categorical data without losing any information.

**For the obtained dataset of shape (107442, 13), Feature Engineering Steps as follows,**

This data set contains temperature points at known distances from a 150 mm by 150 mm heater mat. For use in detecting interlaminate delaminations within a Carbon Fibre Resin Polymer panel.

Temp Change ∘ C = The change in degrees Celsius from the starting temperature of the sample.

x\_heat\_dist = The distance in mm from the centre of the 150 mm by 150 mm square heater mat in the x axis.

y\_heat\_dist = The distance in mm from the centre of the 150 mm by 150 mm square heater mat in the y axis.

defect: If a defect is present: '0' - no defect; '1' - delamination, outer area; '2' - delamination, centre, '3' - no defect with defect between centre of heater mat and probe.

Time: time in seconds from start.

Importing the dataset:

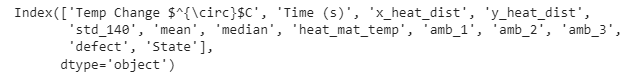
Data = pd.read\_csv("/content/all\_features\_140\_balanced.csv")

Checking data:

Data.head()

Get Column names:

Data.columns

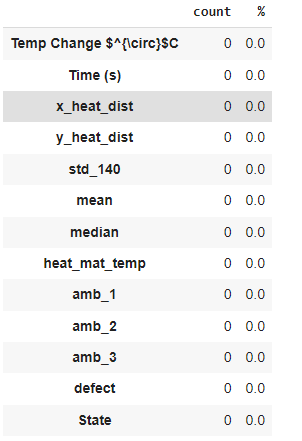


Handling Null values if any:

missing= pd.concat([pd.isnull(Data).sum(), 100 \* pd.isnull(Data).mean()], axis=1)

missing.columns=['count', '%']

missing.sort\_values(by=['count'])

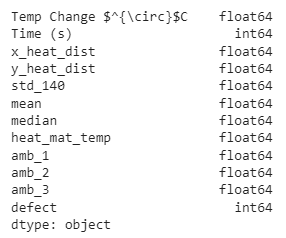


As feature 'State' refers the sematic abbreviation of feature 'defect', we can ignore it.

Data = Data.drop('State', 1)

Checking final datatypes:

Data.dtypes



Checking Target Features

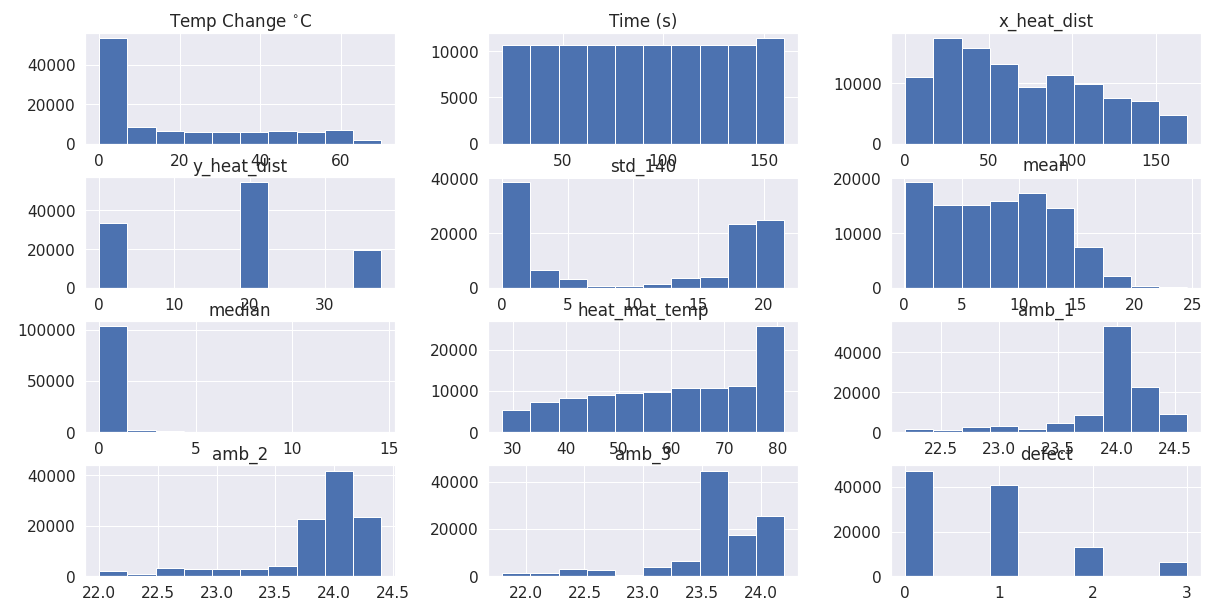
Data.defect.unique()

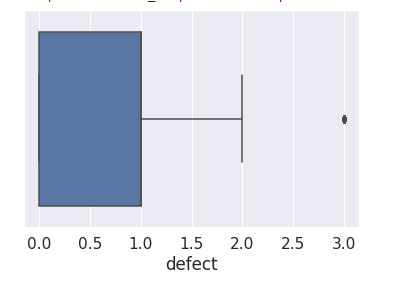
Data['defect'].value\_counts()

Describe the data:

Data.describe().T

Doing Histogram plot:





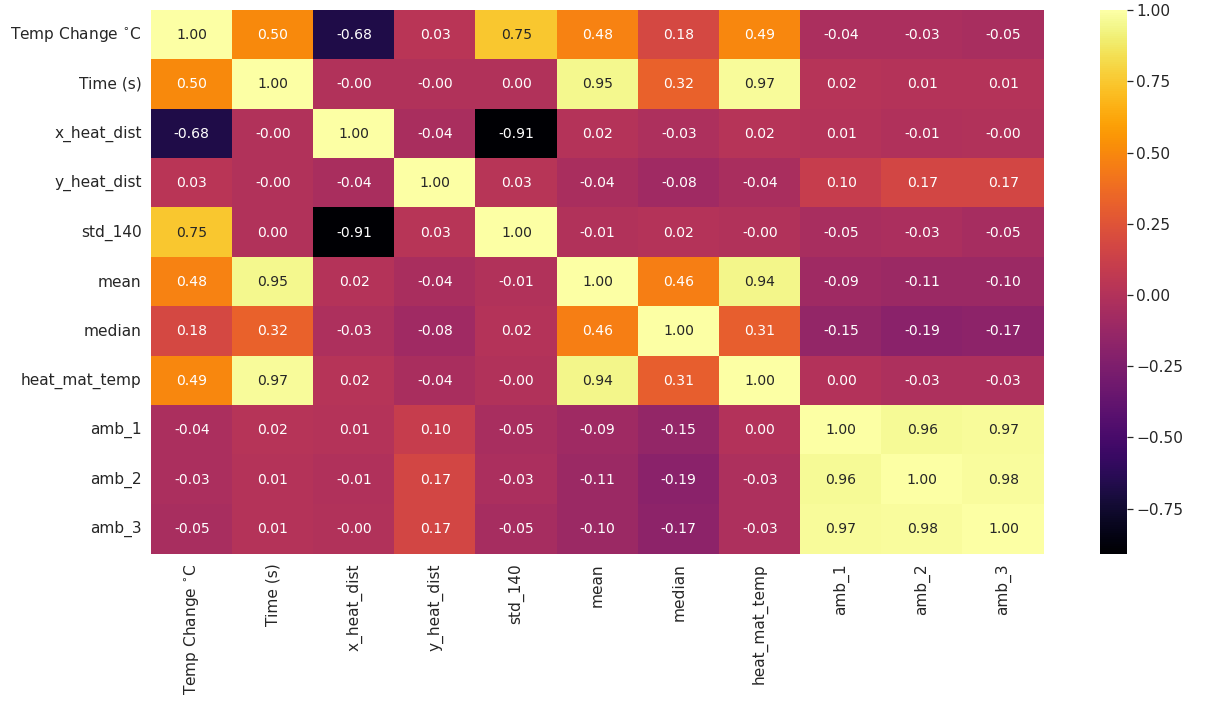
Inference is to keep all the features as it is originating from base x plane.

Doing Heat map to reduce features:

plt.figure(figsize=(20,10))

sns.set(font\_scale=1.4)

sns.heatmap(X.corr(),annot=True, cmap='inferno', fmt='.2f', annot\_kws={'size':14});



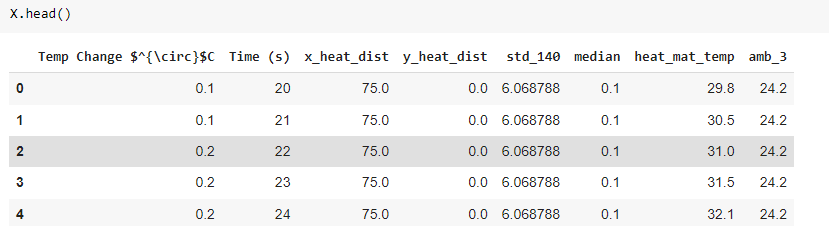
1.As High Correlation occurs between amb\_1, amb\_2, amb\_3 , it is recommended to keep any one among three and which is amb\_3

2.Even though Time(s) and heat\_mat\_temp is highly correlated, Keeping both features supports the context of experiment.

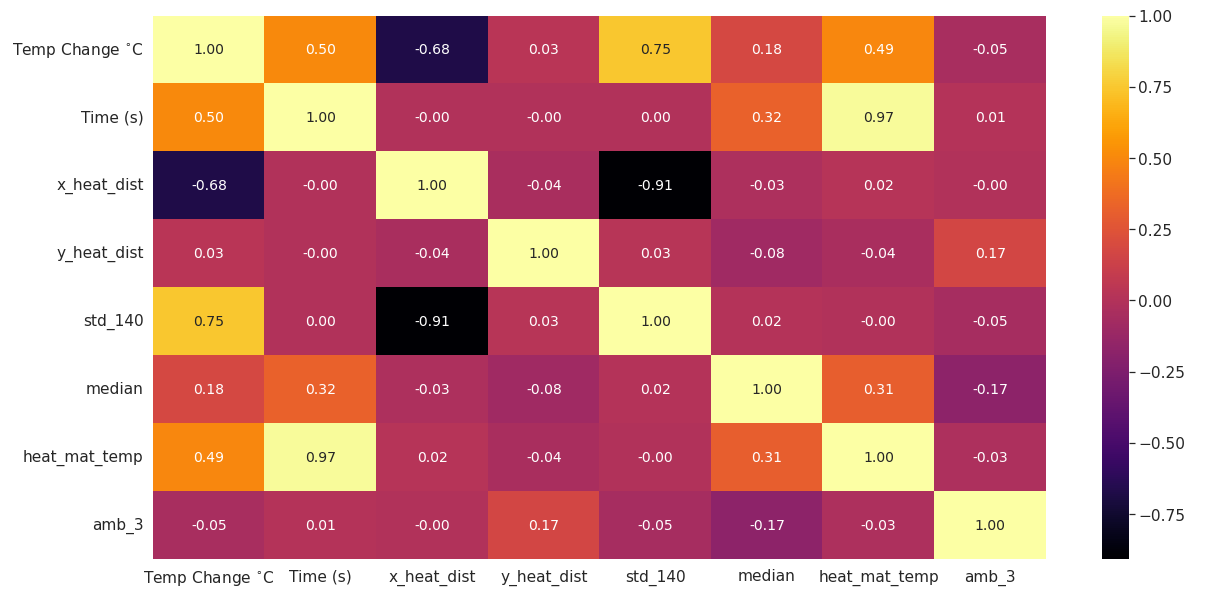
3.mean , time(s) , heat\_mat\_temp is highly correlated and mean is eliminated here as it is derived from other features.

After removing such features,

.drop(columns = ['amb\_1','amb\_2','mean'])



And the new Heat map,



1. **Introduction to Machine Learning:**

Machine Learning tutorial provides basic and advanced concepts of machine learning. Our machine learning tutorial is designed for students and working professionals.

Machine learning is a growing technology which enables computers to learn automatically from past data. Machine learning uses various algorithms for building mathematical models and making predictions using historical data or information. Currently, it is being used for various tasks such as image recognition, speech recognition, email filtering, Facebook auto-tagging, recommender system, and many more.

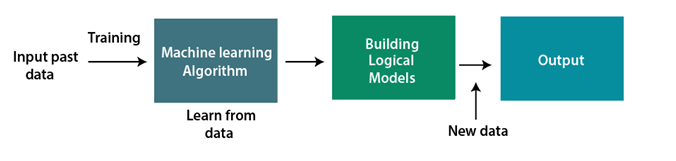
This machine learning tutorial gives you an introduction to machine learning along with the wide range of machine learning techniques such as Supervised, Unsupervised, and Reinforcement learning. You will learn about regression and classification models, clustering methods, hidden Markov models, and various sequential models.

Machine Learning is said as a subset of **artificial intelligence** that is mainly concerned with the development of algorithms which allow a computer to learn from the data and past experiences on their own. The term machine learning was first introduced by **Arthur Samuel** in **1959**. We can define it in a summarized way as:

With the help of sample historical data, which is known as **training data**, machine learning algorithms build a **mathematical model** that helps in making predictions or decisions without being explicitly programmed. Machine learning brings computer science and statistics together for creating predictive models. Machine learning constructs or uses the algorithms that learn from historical data. The more we will provide the information, the higher will be the performance.

A Machine Learning system learns from historical data, builds the prediction models, and whenever it receives new data, predicts the output for it. The accuracy of predicted output depends upon the amount of data, as the huge amount of data helps to build a better model which predicts the output more accurately.

Suppose we have a complex problem, where we need to perform some predictions, so instead of writing a code for it, we just need to feed the data to generic algorithms, and with the help of these algorithms, machine builds the logic as per the data and predict the output. Machine learning has changed our way of thinking about the problem. The below block diagram explains the working of Machine Learning algorithm.



## Features of Machine Learning:

* Machine learning uses data to detect various patterns in a given dataset.
* It can learn from past data and improve automatically.
* It is a data-driven technology.
* Machine learning is much similar to data mining as it also deals with the huge amount of the data.

## Need for Machine Learning

The need for machine learning is increasing day by day. The reason behind the need for machine learning is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannot access the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us.

We can train machine learning algorithms by providing them the huge amount of data and let them explore the data, construct the models, and predict the required output automatically. The performance of the machine learning algorithm depends on the amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money.

The importance of machine learning can be easily understood by its uses cases, Currently, machine learning is used in **self-driving cars**, **cyber fraud detection**, **face recognition**, and **friend suggestion by Facebook**, etc. Various top companies such as Netflix and Amazon have built machine learning models that are using a vast amount of data to analyse the user interest and recommend product accordingly.

**Following are some key points which show the importance of Machine Learning:**

* Rapid increment in the production of data
* Solving complex problems, which are difficult for a human
* Decision making in various sector including finance
* Finding hidden patterns and extracting useful information from data.

## Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

1. **Supervised learning**
2. **Unsupervised learning**
3. **Reinforcement learning**

### 1) Supervised Learning

Supervised learning is a type of machine learning method in which we provide sample labelled data to the machine learning system in order to train it, and on that basis, it predicts the output.

The system creates a model using labelled data to understand the datasets and learn about each data, once the training and processing are done then we test the model by providing a sample data to check whether it is predicting the exact output or not.

The goal of supervised learning is to map input data with the output data. The supervised learning is based on supervision, and it is the same as when a student learns things in the supervision of the teacher. The example of supervised learning is **spam filtering**.

Supervised learning can be grouped further in two categories of algorithms:

* Classification
* Regression

### 2) Unsupervised Learning

Unsupervised learning is a learning method in which a machine learns without any supervision.

The training is provided to the machine with the set of data that has not been labeled, classified, or categorized, and the algorithm needs to act on that data without any supervision. The goal of unsupervised learning is to restructure the input data into new features or a group of objects with similar patterns.

In unsupervised learning, we don't have a predetermined result. The machine tries to find useful insights from the huge amount of data. It can be further classifieds into two categories of algorithms:

* Clustering
* Association

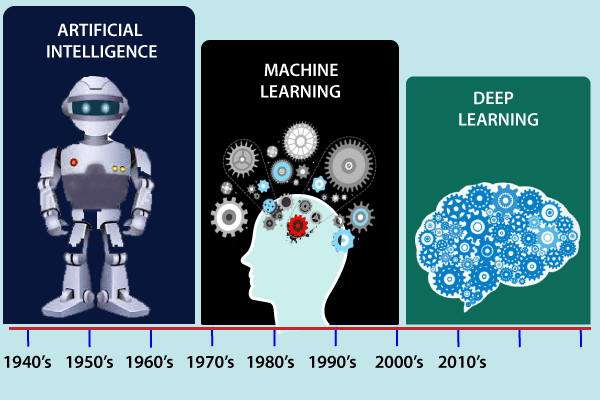
### 3) Reinforcement Learning

Reinforcement learning is a feedback-based learning method, in which a learning agent gets a reward for each right action and gets a penalty for each wrong action. The agent learns automatically with these feedbacks and improves its performance. In reinforcement learning, the agent interacts with the environment and explores it. The goal of an agent is to get the most reward points, and hence, it improves its performance.

The robotic dog, which automatically learns the movement of his arms, is an example of Reinforcement learning.

## History of Machine Learning

Before some years (about 40-50 years), machine learning was science fiction, but today it is the part of our daily life. Machine learning is making our day to day life easy from **self-driving cars** to **Amazon virtual assistant "Alexa"**. However, the idea behind machine learning is so old and has a long history. Below some milestones are given which have occurred in the history of machine learning:



## The early history of Machine Learning (Pre-1940):

* **1834:** In 1834, Charles Babbage, the father of the computer, conceived a device that could be programmed with punch cards. However, the machine was never built, but all modern computers rely on its logical structure.
* **1936:** In 1936, Alan Turing gave a theory that how a machine can determine and execute a set of instructions.

## The era of stored program computers:

* **1940:** In 1940, the first manually operated computer, "ENIAC" was invented, which was the first electronic general-purpose computer. After that stored program computer such as EDSAC in 1949 and EDVAC in 1951 were invented.
* **1943:** In 1943, a human neural network was modeled with an electrical circuit. In 1950, the scientists started applying their idea to work and analyzed how human neurons might work.

## Computer machinery and intelligence:

* **1950:** In 1950, Alan Turing published a seminal paper, "**Computer Machinery and Intelligence**," on the topic of artificial intelligence. **In his paper, he asked, "Can machines think?"**

## Machine intelligence in Games:

* **1952:** Arthur Samuel, who was the pioneer of machine learning, created a program that helped an IBM computer to play a checkers game. It performed better more it played.
* **1959:** In 1959, the term "Machine Learning" was first coined by **Arthur Samuel**.

## The first "AI" winter:

* The duration of 1974 to 1980 was the tough time for AI and ML researchers, and this duration was called as **AI winter**.
* In this duration, failure of machine translation occurred, and people had reduced their interest from AI, which led to reduced funding by the government to the researches.

## Machine Learning from theory to reality

* **1959:** In 1959, the first neural network was applied to a real-world problem to remove echoes over phone lines using an adaptive filter.
* **1985:** In 1985, Terry Sejnowski and Charles Rosenberg invented a neural network **NETtalk**, which was able to teach itself how to correctly pronounce 20,000 words in one week.
* **1997:** The IBM's **Deep blue** intelligent computer won the chess game against the chess expert Garry Kasparov, and it became the first computer which had beaten a human chess expert.

## Machine Learning at 21st century

* **2006:** In the year 2006, computer scientist Geoffrey Hinton has given a new name to neural net research as "**deep learning**," and nowadays, it has become one of the most trending technologies.
* **2012:** In 2012, Google created a deep neural network which learned to recognize the image of humans and cats in YouTube videos.
* **2014:** In 2014, the Chabot "**Eugen Goostman**" cleared the Turing Test. It was the first Chabot who convinced the 33% of human judges that it was not a machine.
* **2014:** **DeepFace** was a deep neural network created by Facebook, and they claimed that it could recognize a person with the same precision as a human can do.
* **2016:** **AlphaGo** beat the world's number second player **Lee sedol** at **Go game**. In 2017 it beat the number one player of this game **Ke Jie**.
* **2017:** In 2017, the Alphabet's Jigsaw team built an intelligent system that was able to learn the **online trolling**. It used to read millions of comments of different websites to learn to stop online trolling.

## Machine Learning at present:

Now machine learning has got a great advancement in its research, and it is present everywhere around us, such as **self-driving cars**, **Amazon Alexa**, **Catboats**, **recommender system**, and many more. It includes **Supervised**, **unsupervised**, and **reinforcement learning with clustering**, **classification**, **decision tree**, **SVM algorithms**, etc.

Modern machine learning models can be used for making various predictions, including **weather prediction**, **disease prediction**, **stock market analysis**, etc.

## Prerequisites

Before learning machine learning, you must have the basic knowledge of followings so that you can easily understand the concepts of machine learning:

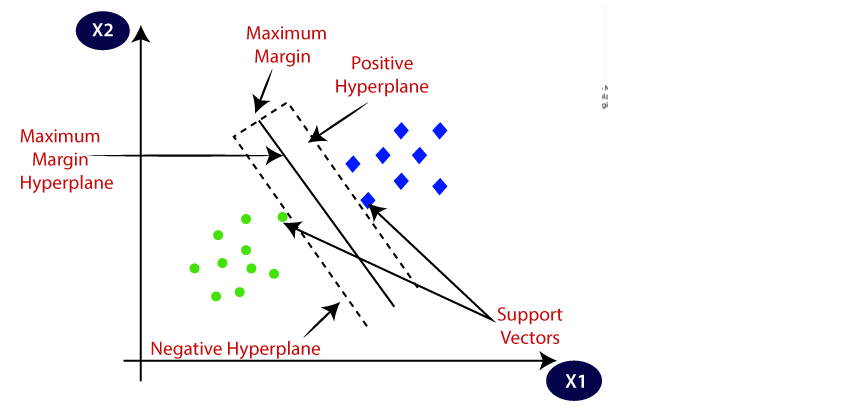
* Fundamental knowledge of probability and linear algebra.
* The ability to code in any computer language, especially in Python language.
* Knowledge of Calculus, especially derivatives of single variable and multivariate functions.

# Support Vector Machine Algorithm

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.

## Types of SVM

**SVM can be of two types:**

* **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
* **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

## Hyperplane and Support Vectors in the SVM algorithm:

**Hyperplane:** There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.

The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.

We always create a hyperplane that has a maximum margin, which means the maximum distance between the data points.

**Support Vectors:**

The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector. Since these vectors support the hyperplane, hence called a Support vector.

## How does SVM works?

**Linear SVM:**

The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:



So as it is 2-d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes. Consider the below image:



Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.



Non-Linear SVM:

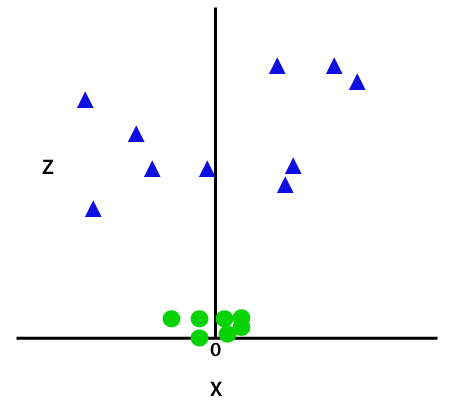
If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:



So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third dimension z. It can be calculated as:

z=x2 +y2

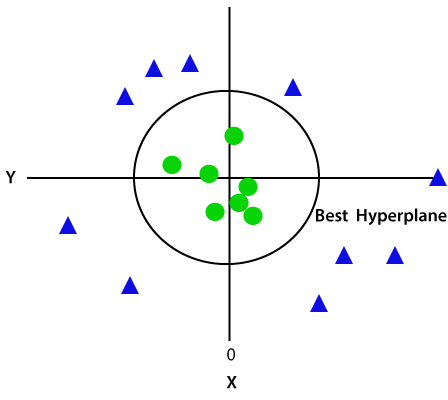
By adding the third dimension, the sample space will become as below image:



So now, SVM will divide the datasets into classes in the following way. Consider the below image:



Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



Hence, we get a circumference of radius 1 in case of non-linear data.

## **Introduction to Deep learning Algorithms:**

Deep learning can be defined as the method of machine learning and artificial intelligence that is intended to intimidate humans and their actions based on certain human brain functions to make effective decisions. It is a very important data science element that channels its modeling based on data-driven techniques under**predictive modeling** and **statistics.** To drive such a human-like ability to adapt and learn and to function accordingly, there have to be some strong forces which we popularly called **algorithms.**

[Deep learning](https://www.javatpoint.com/deep-learning)

algorithms are dynamically made to run through several **layers** of neural networks, which are nothing but a set of decision-making networks that are pre-trained to serve a task. Later, each of these is passed through simple layered representations and move on to the next layer. However, most [machine learning](https://www.javatpoint.com/machine-learning)

is trained to work fairly well on datasets that have to deal with hundreds of features or columns. For a data set to be structured or unstructured, machine learning tends to fail mostly because they fail to recognize a simple image having a dimension of **800x1000** in RGB. It becomes quite unfeasible for a traditional machine learning algorithm to handle such depths. This is where deep learning.

## Importance of Deep Learning

Deep learning algorithms play a crucial role in determining the features and can handle the large number of processes for the data that might be structured or unstructured. Although, deep learning algorithms can overkill some tasks that might involve complex problems because they need access to huge amounts of data so that they can function effectively. For example, there's a popular deep learning tool that recognizes images namely **Imagenet** that has access to **14 million** images in its dataset-driven algorithms. It is a highly comprehensive tool that has defined a next-level benchmark for deep learning tools that aim images as their dataset.

Deep learning algorithms are highly progressive algorithms that learn about the image that we discussed previously by passing it through each neural network layer. The layers are highly sensitive to detect low-level features of the image like **edges** and **pixels** and henceforth the combined layers take this information and form holistic representations by comparing it with previous data. For example, the middle layer might be programmed to detect some special parts of the object in the photograph which other deep trained layers are programmed to detect special objects like **dogs, trees, utensils,** etc.

However, if we talk out the simple task that involves less complexity and a data-driven resource, deep learning algorithms fail to generalize simple data. This is one of the main reasons deep learning is not considered effective as **linear** or **boosted tree models.** Simple models aim to churn out custom data, track fraudulent transactions and deal with less complex datasets with fewer features. Also, there are various cases like **multiclass classification** where deep learning can be effective because it involves smaller but more structured datasets but is not preferred usually.

Having said that, let's look understand some of the most important deep learning algorithms given below.

## Deep Learning Algorithms

The Deep Learning Algorithms are as follows:

### 1. Convolutional Neural Networks (CNNs)

[CNN's](https://www.javatpoint.com/keras-convolutional-neural-network)

popularly known as **ConvNets** majorly consists of several layers and are specifically used for image processing and detection of objects. It was developed in **1998** by **Yann LeCun** and was first called **LeNet.** Back then, it was developed to recognize digits and zip code characters. CNNs have wide usage in identifying the image of the satellites, medical image processing, series forecasting, and anomaly detection.

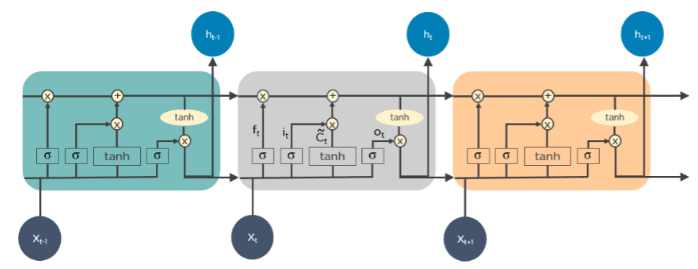
CNNs process the data by passing it through multiple layers and extracting features to exhibit convolutional operations. The **Convolutional Layer** consists of **Rectified Linear Unit** (ReLU) that outlasts to rectify the feature map. **The Pooling layer** is used to rectify these feature maps into the next feed. Pooling is generally a sampling algorithm that is down-sampled and it reduces the dimensions of the feature map. Later, the result generated consists of **2-D arrays** consisting of **single, long, continuous,** and **linear vector** flattened in the map. The next layer i.e., called **Fully Connected Layer** which forms the flattened **matrix** or **2-D** array fetched from the Pooling Layer as input and identifies the image by classifying it.

### Deep Learning Algorithms 2. Long Short-Term Memory Networks (LSTMs)

[LSTMs](https://www.javatpoint.com/long-short-term-memory-rnn-in-tensorflow)

can be defined as **Recurrent Neural Networks** (RNN) that are programmed to learn and adapt for dependencies for the long term. It can memorize and recall past data for a greater period and by default, it is its sole behavior. LSTMs are designed to retain over time and henceforth they are majorly used in time series predictions because they can restrain memory or previous inputs. This analogy comes from their **chain-like** structure consisting of **four** interacting layers that communicate with each other differently. Besides applications of time series prediction, they can be used to construct **speech recognizers, development in pharmaceuticals,** and composition of **music loops** as well.

LSTM work in a sequence of events. First, they don't tend to remember irrelevant details attained in the previous state. Next, they update certain cell-state values selectively and finally generate certain parts of the cell-state as output. Below is the diagram of their operation.

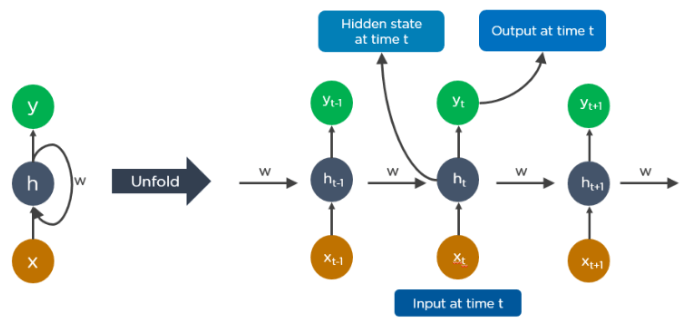


### 3. Recurrent Neural Networks (RNNs)

[Recurrent Neural Networks](https://www.javatpoint.com/keras-recurrent-neural-networks)

or RNNs consist of some directed connections that form a cycle that allow the input provided from the LSTMs to be used as input in the current phase of RNNs. These inputs are deeply embedded as inputs and enforce the memorization ability of LSTMs lets these inputs get absorbed for a period in the internal memory. RNNs are therefore dependent on the inputs that are preserved by LSTMs and work under the synchronization phenomenon of LSTMs. RNNs are mostly used in captioning the image, time series analysis, recognizing handwritten data, and translating data to machines.

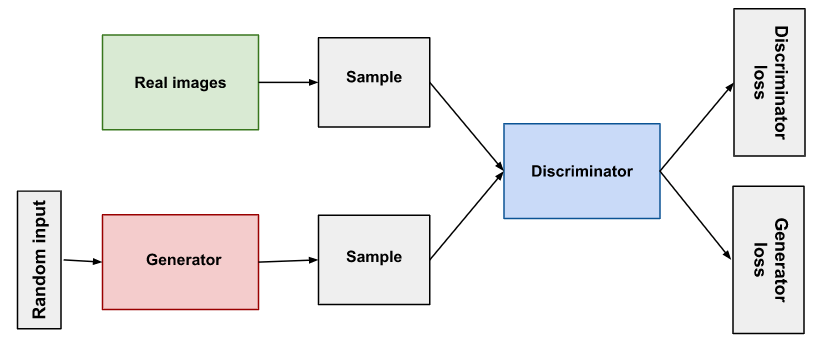
RNNs follow the work approach by putting output feeds **(t-1)** time if the time is defined as **t.** Next, the output determined by t is feed at input time **t+1.** Similarly, these processes are repeated for all the input consisting of any length. There's also a fact about RNNs is that they store historical information and there's no increase in the input size even if the model size is increased. RNNs look something like this when unfolded.



### 4. Generative Adversarial Networks (GANs)

GANs are defined as deep learning algorithms that are used to generate new instances of data that match the training data. GAN usually consists of two components namely a **generator** that learns to generate false data and a **discriminator** that adapts itself by learning from this false data. Over some time, GANs have gained immense usage since they are frequently being used to clarify **astronomical images** and simulate **lensing** the gravitational dark matter. It is also used in **video games** to increase graphics for **2D** textures by recreating them in higher resolution like **4K**. They are also used in creating **realistic cartoons character** and also rendering human faces and **3D object rendering.**

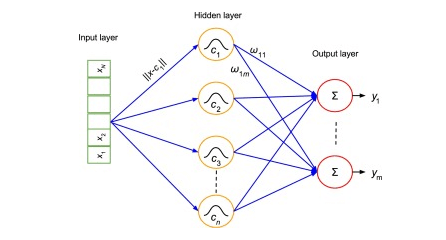
GANs work in simulation by generating and understanding the fake data and the real data. During the training to understand these data, the generator produces different kinds of fake data where the discriminator quickly learns to adapt and respond to it as false data. GANs then send these recognized results for updating. Consider the below image to visualize the functioning.



### 5. Radial Basis Function Networks (RBFNs)

RBFNs are specific types of neural networks that follow a feed-forward approach and make use of radial functions as activation functions. They consist of **three** layers namely the **input layer, hidden layer,** and **output layer** which are mostly used for **time-series prediction, regression testing,** and **classification.**

RBFNs do these tasks by measuring the similarities present in the training data set. They usually have an input vector that feeds these data into the input layer thereby confirming the identification and rolling out results by comparing previous data sets. Precisely, the input layer has **neurons** that are sensitive to these data and the nodes in the layer are efficient in classifying the class of data. Neurons are originally present in the hidden layer though they work in close integration with the input layer. The hidden layer contains **Gaussian transfer** functions that are inversely proportional to the distance of the output from the neuron's center. The output layer has linear combinations of the **radial-based** data where the Gaussian functions are passed in the neuron as parameter and output is generated. Consiider the given image below to understand the process thoroughly.

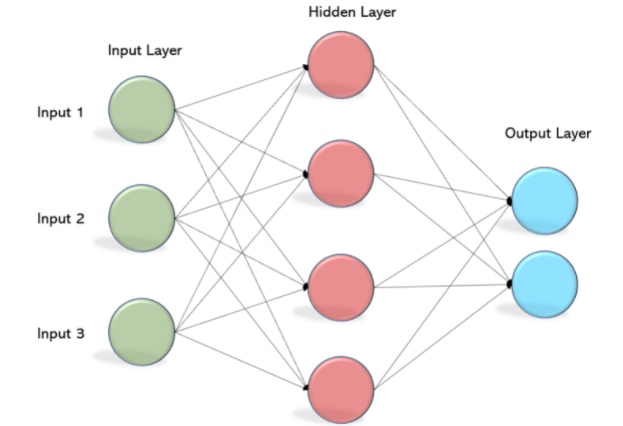


### 6. Multilayer Perceptrons (MLPs)

[MLPs](https://www.javatpoint.com/multi-layer-perceptron-in-tensorflow)

are the base of deep learning technology. It belongs to a class of feed-forward neural networks having various layers of **perceptrons.** These perceptrons have various activation functions in them. MLPs also have connected input and output layers and their number is the same. Also, there's a layer that remains hidden amidst these two layers. MLPs are mostly used to build **image and speech recognition** systems or some other types of the **translation software.**

The working of MLPs starts by feeding the data in the input layer. The neurons present in the layer form a graph to establish a connection that passes in one direction. The weight of this input data is found to exist between the hidden layer and the input layer. MLPs use activation functions to determine which nodes are ready to fire. These activation functions include **tanh** function, **sigmoid** and **ReLUs.** MLPs are mainly used to train the models to understand what kind of co-relation the layers are serving to achieve the desired output from the given data set. See the below image to understand better.

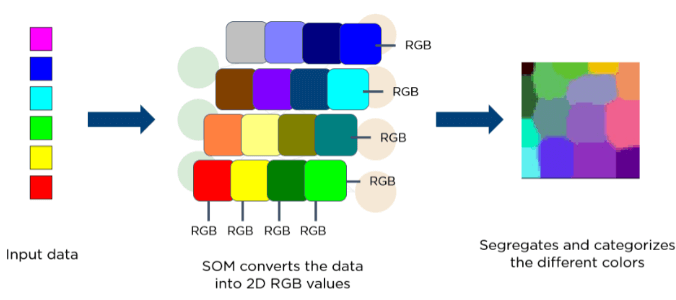


### 7. Self Organizing Maps (SOMs)

[SOMs](https://www.javatpoint.com/keras-kohonen-self-organizing-maps)

were invented by **Teuvo Kohenen** for achieving data visualization to understand the dimensions of data through artificial and self-organizing neural networks. The attempts to achieve data visualization to solve problems are mainly done by what humans cannot visualize. These data are generally high-dimensional so there are lesser chances of human involvement and of course less error.

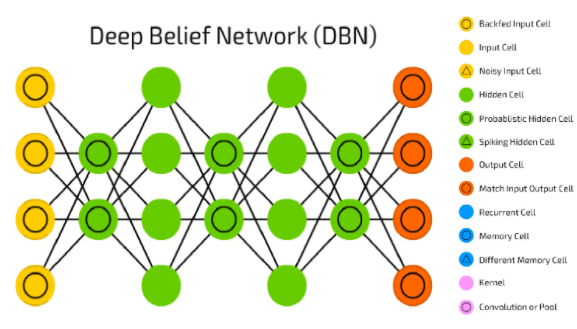
SOMs help in visualizing the data by initializing weights of different nodes and then choose random vectors from the given training data. They examine each node to find the relative weights so that dependencies can be understood. The winning node is decided and that is called **Best Matching Unit** (BMU). Later, SOMs discover these winning nodes but the nodes reduce over time from the sample vector. So, the closer the node to BMU more is the more chance to recognize the weight and carry out further activities. There are also multiple iterations done to ensure that no node closer to BMU is missed. One example of such is the **RGB color combinations** that we use in our daily tasks. Consider the below image to understand how they function.



### 8. Deep Belief Networks (DBNs)

DBNs are called generative models because they have various layers of latent as well as stochastic variables. The latent variable is called a **hidden unit** because they have binary values. DBNs are also called **Boltzmann Machines** because the **RGM** layers are stacked over each other to establish communication with previous and consecutive layers. DBNs are used in applications like video and image recognition as well as capturing motional objects.

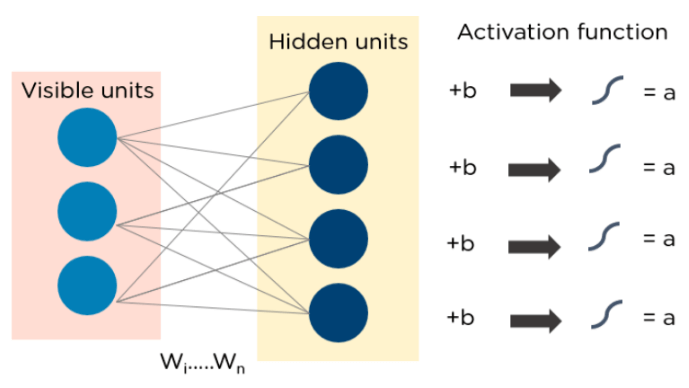
DBNs are powered by **Greedy algorithms.** The layer to layer approach by leaning through a **top-down** approach to generate weights is the most common way DBNs function. DBNs use step by step approach of **Gibbs** sampling on the hidden **two-layer** at the top. Then, these stages draw a sample from the visible units using a model that follows the ancestral sampling method. DBNs learn from the values present in the latent value from every layer following the **bottom-up** pass approach.



### 9. Restricted Boltzmann Machines (RBMs)

RBMs were developed by **Geoffrey Hinton** and resemble stochastic neural networks that learn from the probability distribution in the given input set. This algorithm is mainly used in the field of dimension **reduction, regression** and **classification, topic modeling** and are considered the building blocks of DBNs. RBIs consist of two layers namely the **visible layer** and the **hidden layer**. Both of these layers are connected through hidden units and have bias units connected to nodes that generate the output. Usually, RBMs have two phases namely **forward pass** and **backward pass**.

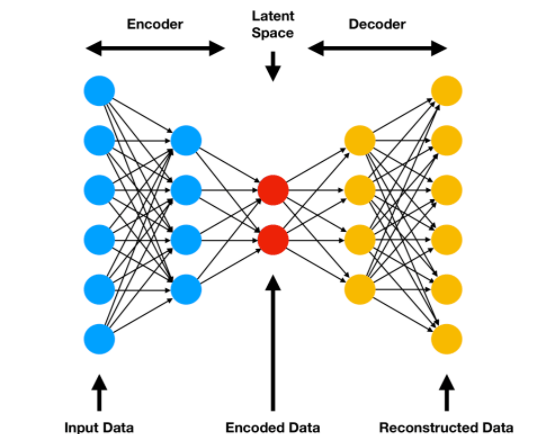
The functioning of RBMs is carried out by accepting inputs and translating them to numbers so that inputs are encoded in the forward pass. RBMs take into account the weight of every input, and the backward pass takes these input weights and translates them further into reconstructed inputs. Later, both of these translated inputs, along with individual weights, are combined. These inputs are then pushed to the visible layer where the activation is carried out, and output is generated that can be easily reconstructed. To understand this process, consider the below image.



### Autoencoders

Autoencoders are a special type of neural network where inputs are outputs are found usually identical. It was designed to primarily solve the problems related to unsupervised learning. Autoencoders are highly trained neural networks that **replicate** the data. It is the reason why the input and output are generally the same. They are used to achieve tasks like **pharma discovery, image processing,** and **population prediction.**

Autoencoders constitute three components namely the **encoder**, the **code**, and the **decoder.** Autoencoders are built in such a structure that they can receive inputs and transform them into various representations. The attempts to copy the original input by reconstructing them is more accurate. They do this by encoding the image or input, reduce the size. If the image is not visible properly they are passed to the neural network for clarification. Then, the clarified image is termed a reconstructed image and this resembles as accurate as of the previous image. To understand this complex process, see the below-provided image.



1. **Result and Discussion:**

**Implementation of Machine learning and Deep Learning Models:**

Data Modelling Using Machine Learning Algorithm (Support Vector Machining)

Importing dependencies:

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

#Split data into features and target.

y = Data['defect']

X = Data.drop(columns = 'defect')

Importing Flattener and splitting of data:

SS\_scaler = StandardScaler()

# Fit the standard scaler to the data.

x\_std = SS\_scaler.fit\_transform(X)

#Create training and test data.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

    x\_std, y, test\_size=0.2, random\_state=42)

Import the Support Vector Machine and Fit the data:

from sklearn import svm

svc = svm.SVC(kernel='rbf', C=0.1,gamma=10).fit(X\_train, y\_train)

y\_pred = svc.predict(X\_test)

Checking Performance of the Model:

from sklearn.metrics import accuracy\_score,multilabel\_confusion\_matrix

print("svm model accuracy(in %):", accuracy\_score(y\_test, y\_pred)\*100)

Hyper parameter tuning:

from sklearn.svm import SVC

from sklearn.model\_selection import GridSearchCV

param\_grid = {'C': [0.01, 0.1, 1, 10, 100],

              'gamma': [1, 0.1, 0.01, 0.001, 0.0001],

              'kernel': ['linear','rbf','poly','sigmoid']}

grid = GridSearchCV(SVC(), param\_grid, refit = True, verbose = 3)

grid.fit(X\_train, y\_train)

Saving machine Learning Model:

import pickle

pickle.dump(svc, open('svc\_model.pkl','wb'))

Data Modelling using Deep Neural Network:

Importing dependencies:

import tensorflow

from tensorflow.keras.models import Sequential

# Sequential model relates to forward and back propagation

from tensorflow.keras.layers import Dense

# Dense layer relates to I/O,HL,O/T layer

from tensorflow.keras.layers import ReLU

# import Activation functions

from tensorflow.keras.layers import Dropout

# import dropout layer

Import the ANN and Fit the data:

Classifier = Sequential()

#Assigning classifier as sequential neural network model

Classifier.add(Dense(units = 8, activation = 'relu'))

#Assigning Input layer as Dense layer

Classifier.add(Dense(units = 100, activation = 'relu'))

#Assigning Hidden Layer 1

Classifier.add(Dropout(0.3))

#Addidng dropout layer

Classifier.add(Dense(units = 100, activation = 'relu'))

#Assigning Hidden layer 2

Classifier.add(Dense(units = 4, activation = 'softmax'))

 #Assigning output layer

#Compiling ANN

Optimizer = tensorflow.keras.optimizers.Adam(learning\_rate=0.001)

Classifier.compile(optimizer=Optimizer,loss='sparse\_categorical\_crossentropy', metrics=['accuracy'])

#Classifier is ready to be trained and configured with learning\_rate, cost\_function and metrics

#Early\_Stopping:

import tensorflow as tf

early\_stopping = tf.keras.callbacks.EarlyStopping(monitor='val\_loss',min\_delta=0.0001,patience = 20,verbose=1,mode='auto',baseline=None,restore\_best\_weights=False)

ANN\_model = Classifier.fit(X\_train,y\_train,validation\_split=0.33,batch\_size=10, epochs=100, callbacks = early\_stopping)

Checking Performance of the Model:

from sklearn.metrics import multilabel\_confusion\_matrix,accuracy\_score

cm = multilabel\_confusion\_matrix(y\_test,y\_pred\_class)

accuracy\_score = accuracy\_score(y\_test,y\_pred\_class)

Saving Deep Learning Model:

Classifier.save('Model.h5')

**Compare and Contrast ML and DL models and its performance:**

|  |  |
| --- | --- |
| Machine learning Model | Deep Learning Model |
| Accuracy: 88% | Accuracy: 94% |
| Low Complexity | High Complexity |
| Low performed metrics | High performed metrics |
| Sklearn Framework used | TensorFlow framework is used |
| Saved through .pkl format | Saved through .h5 format |
| Low Inference potential on defect analysis | High Inference potential on defect analysis. |

#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_#\_

The End

References:

1. Gillespie, D.I.; Hamilton, A.W.; Kruckenberg, T.; Neilson, B.; Atkinson, R.C.; Andonovic, I.; Tachtatzis, C.

Geometrical Thermal Analysis as a Form of Finite Element Analysis Enhancement. In Advances

in Manufacturing Technology XXXIII, Proceedings of the 17th International Conference on Manufacturing Research,

Incorporating the 34th National Conference on Manufacturing Research, 10–12 September 2019; Jin, Y.M.P., Ed.;

Queen’s University: Belfast, UK, 2019; pp. 187–192. [CrossRef]

2. Gillespie, D.I.; Hamilton, A.W.; McKay, E.J.; Neilson, B.; Atkinson, R.C.; Andonovic, I.; Tachtatzis, C.

Non-destructive identification of fibre orientation in multi-ply biaxial laminates using contact temperature

sensors. Sensors **2020**, 20, 3865.[CrossRef] [PubMed]

3. Gillespie, D.I.; Hamilton, A.W.; Atkinson, R.C.; Bellekens, X.; Michie, C.; Andonovic, I.; Tachtatzis, C.

Defect Detection in Aerospace Sandwich Composite Panels Using Conductive Thermography and Contact

Sensors. Sensors **2020**, 20, 6689. [CrossRef] [PubMed]

4. Mills, J.A.; Hamilton, A.W.; Gillespie, D.I.; Andonovic, I.; Michie, C.; Burnham, K.; Tachtatzis, C.

Identifying Defects in Aerospace Composite Sandwich Panels Using High-Definition Distributed Optical

Fibre Sensors. Sensors **2020**, 20, 6746. [CrossRef] [PubMed]

5. Delaney, M.P.; Fung, S.Y.; Kim, H. Dent depth visibility versus delamination damage for impact of composite

panels by tips of varying radius. J. Compos. Mater. **2018**, 52, 2691–2705. [CrossRef]

6. Hongkarnjanakul, N.; Bouvet, C.; Rivallant, S. Validation of low velocity impact modelling on different

stacking sequences of CFRP laminates and influence of fibre failure. Compos. Struct. **2013**, 106, 549–559.

[CrossRef]

7. Long, S.; Yao, X.; Zhang, X. Delamination prediction in composite laminates under low-velocity impact.

Compos. Struct. **2015**, 132, 290–298. [CrossRef]

8. Singh, H.; Namala, K.K.; Mahajan, P. A damage evolution study of E-glass/epoxy composite under low

velocity impact. Compos. Part B Eng. **2015**, 76, 235–248. [CrossRef]

9. Scarponi, C.; Briotti, G. Ultrasonic technique for the evaluation of delaminations on CFRP, GFRP,

KFRP composite materials. Compos. Part B Eng. **2000**, 31, 237–243. [CrossRef]

10. Benammar, A.; Drai, R.; Guessoum, A. Detection of delamination defects in CFRP materials using ultrasonic

signal processing. Ultrasonics **2008**, 48, 731–738. [CrossRef]

11. Zhang, Z.; Guo, S.; Li, Q.; Cui, F.; Malcolm, A.A.; Su, Z.; Liu, M. Ultrasonic detection and characterization

of delamination and rich resin in thick composites with waviness. Compos. Sci. Technol. **2020**, 189, 108016.

[CrossRef]

12. Meola, C.; Boccardi, S.; Carlomagno, G.; Boffa, N.; Monaco, E.; Ricci, F. Nondestructive evaluation of carbon

fibre reinforced composites with infrared thermography and ultrasonics. Compos. Struct. **2015**, 134, 845–853.

[CrossRef]

13. Cheng, L.; Tian, G.Y. Comparison of nondestructive testing methods on detection of delaminations in

composites. J. Sens. **2012**, 2012. [CrossRef]

14. Wang, Q.; Hu, Q.; Qiu, J.; Pei, C.; Li, X.; Zhou, H. Using differential spread laser infrared thermography to

detect delamination and impact damage in CFRP. Infrared Phys. Technol. **2020**, 106, 103282. [CrossRef]

15. Moskovchenko, A.I.; Vavilov, V.P.; Bernegger, R.; Maierhofer, C.; Chulkov, A.O. Detecting Delaminations

in Semitransparent Glass Fiber Composite by Using Pulsed Infrared Thermography. J. Nondestr. Evaluat.

**2020**, 39, 69. [CrossRef]

16. Mondal, S.; Ramachandra, L.S. Stability and failure analyses of delaminated composite plates subjected

to localized heating. Compos. Struct. **2019**, 209, 258–267. [CrossRef]

17. Kharghani, N.; Guedes Soares, C. Behavior of composite laminates with embedded delaminations.

Compos. Struct. **2016**, 150, 226–239. [CrossRef]

18. Civil Aviation Authority. Safety and Airspace Regulation Group Mandatory Requirements for Airworthiness;

Civil Aviation Authority: London, UK, 2017.

19. UK National Aerospace NDT Board. UK NANDTB 18 NDT Method or Technique-Training and Certification

Guidance; UK National Aerospace NDT Board: Northampton, UK, 2015.

20. Farhat, N.H. Photonit neural networks and learning mathines the role of electron-trapping materials.

IEEE Expert Intell. Syst. Appl. **1992**, 7, 63–72. [CrossRef]

Sensors **2020**, 20, 7227 19 of 19

21. Platt, J.C. A Fast Algorithm for Training Support Vector Machines; Technical Report; Microsoft:

Redmond,WA, USA, 1998.

22. Kulkarni, A.; Kulkarni, N. Fuzzy Neural Network for Pattern Classification. Proc. Comp. Sci. **2020**,

167, 2606–2616. [CrossRef]

23. Li, H.X.; Yang, J.L.; Zhang, G.; Fan, B. Probabilistic support vector machines for classification of noise

affected data. Inf. Sci. **2013**, 221, 60–71. [CrossRef]

24. Ben-Hur, A.; Horn, D.; Siegelmann, H.T.; Vapnik, V. Support Vector Clustering. J. Mach. Learn. Res. **2002**,

2, 125–137. [CrossRef]

25. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.;

Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. J. Mach. Learn. Res. **2011**,

12, 2825–2830.

26. Akbani, R.; Kwek, S.; Japkowicz, N. Applying Support Vector Machines to Imbalanced Datasets.

In Machine Learning: ECML 2004; Boulicaut, J.F., Esposito, F., Giannotti, F., Pedreschi, D., Eds.;

Springer: Berlin/Heidelberg, Germany, 2004; pp. 39–50.