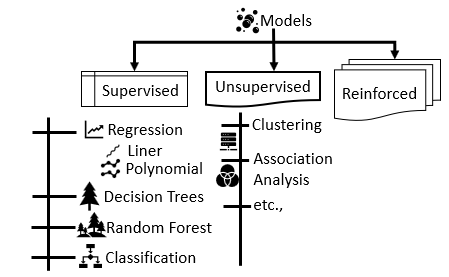
## Introduction

The field of data and algorithms is dynamic, growing and increasingly complex. As the field grows larger and more complex, the need for skilled analysis and practitioners becomes more important towards proving the affinity of the algorithm. The efficacy of the algorithm is proven when the data is infused to validate the logic embedded within it. An algorithm is a sequence of instructions supported by the data to derive a result. The logic embedded within these instruction sequences is the key component that specifies the knowledge to solve problems. But, there has to be a control component that decides which problem-solving strategies to use. The logic component determines the meaning of an algorithm whereas the control component only affects its efficiency.

It is often possible to improve an algorithm's efficiency without changing its logic, but, improving its control component without changing its logic. The method of improving the is achieved with the adaptive nature of the algorithm imbibed within. More often this is referred to as the learning component of the algorithm. Such learning components can be both supervised as well as unsupervised. My study is towards a supervised model and identifying a pattern. Researchers are investing their time & efforts to identify if programmable machines could learn from the inflow of data. In this chapter, I will document the literature research on regression algorithms which is a rapidly-evolving field. I will summarize various research papers, highlighting their key findings along with their potential implications as well.

These algorithms are put into practice by machines to understand the existing data and predict the progression of the data. Thus, the world of machine learning models evolved. There are mainly 3 categories these models fall into by the nature of their functionality. They are mainly either supervised or unsupervised. Although there is another model, Reinforced, mostly used with either the environment or the programming language that is dynamic in nature.



##### Picture 1: Types of Models

Regression algorithms are classified under the supervised learning class of machine learning techniques used to predict continuous target (output) values from the input variables that are expressed with a common set of properties. This common set of properties is considered a dimension. Regression analysis is a statistical method used to predict a continuous outcome variable (also known as the dependent variable) based on one or more predictor variables (also known as independent variables or features). In this chapter, I’m going to articulate the various traditions of these algorithms with high-dimensional data, which has more than one feature or variable. In the case of high-dimensional data, multiple input variables are used to predict the output variable.

High-dimensional data is derived from various sources. The major 3 sources generating data with standard characteristics are transactions, interactions and observations. These sources will have defined properties for each data point. But, the data continuum is not just from these, mostly from other entities like the internet, IoT (or) connected devices, business data feeds, service logs and social interaction feeds.

This data continuum's problem is normalising with a common set of properties. Not only that, the frequency with which these data points are emitted is another challenge to collect and process. The bigger challenge is not only in collecting them and normalizing them but, executing the mean function with different predictor values. For example, for a single characteristic of *p* (a data point) in the data continuum, identified as, *p1, p2, p3, …, pn* where *n* is the count of the data continuum at any given interval. The norm of *p* is ǁ*p*ǁ and mathematically written as

ǁ*p*ǁ = (*p*’*p*)½= () ½

This is a simple definition of (*n X 1*) vector or a single-dimensional data continuum. The complexity increases every known set of characteristics applied to each data point of the continuum.

### Various Regression models for High-dimensional data

Various regression algorithms can be used for high-dimensional data, including linear, polynomial, and support vector regression. Each of these algorithms has its strengths and weaknesses, and the choice of algorithm will depend on the specific problem being solved and the characteristics of the data being used.

#### Liner Regression

Linear regression is a simple and widely used algorithm that assumes a linear relationship between the input variables and the output variable from simple and complex data sets, and it is relatively easy to interpret the results. However, it is not suitable for data sets with non-linear relationships or data sets with a large number of input variables. This is a statistical method used to model the relationship between a single dependent variable, y, and one or more independent variables, x. In the case of multi-dimensional data, linear regression can be extended to include multiple independent variables, represented by the vector x = [x1, x2, ..., xn]. The basic form of linear regression is represented by the following equation:

y = β0 + β1X1 + β2X2 + ... + βnXn + ϵ

where β0, β1, β2, ..., βn are the coefficients of the independent variables, and ϵ is the error term. The purpose of Linear Regression is to derive the values of the coefficients that minimize the sum of the squared errors between the predicted values of y and the actual values of y.

#### Polynomial Regression

Polynomial regression is a type of linear regression that is used when the relationship between the input variables and the output variable is non-linear, and the relationship between the predictor variable(s) and the outcome variable is modelled as an nth degree polynomial equation. This algorithm can be used to model complex data sets and it can handle a large number of input variables. However, it can be sensitive to outliers and it can be difficult to interpret the results. A model for a single predictor, X, can be written as

Y = β0 + β1X + β2X2 + … + βhXh + ϵ

where h is the degree of the polynomial. This deals with a nonlinear relationship between Y and X, yet, is still considered a flavour of linear regression since it is linear in the regression coefficients, β1,β2,...,βh. This type of regression can also be extended to multi-dimensional data by using multiple independent variables in the polynomial equation.

#### Support Vector Regression (SVR)

One of the most powerful regression algorithms for multi-dimensional data is Support Vector Regression, which uses support vectors to find the optimal boundary between the input variables and the output variable. Its ability to handle non-linear relationships between features and target values by transforming the data into high-dimensional space and finding a linear boundary in that space. The core of this algorithm is the kernel mechanism, which transforms the data into a high-dimensional space where it becomes possible to find a linear boundary that separates the classes.

SVR is an extension of the support vector machine (SVM), which is used for classification. The main difference between SVR and SVM is that SVR is used for predicting continuous values, while SVM is used for predicting categorical values.

#### Others and conclusion

In addition to these algorithms, other techniques can be used for multi-dimensional data such as Random Forest, Gradient Boosting, Neural Networks, Lasso, Ridge, etc.

Finally, the performance of regression models must be evaluated using appropriate measuring techniques such as R-squared, Mean Squared Error (MSE) and Root Mean Squared Error (RMSE). These metrics can help determine how well the model predicts the outcome variable based on the predictor variable. In addition, it is important to consider the assumptions of the regression algorithm and check for possible violations of these assumptions, such as the normality of the error and the independence of the error.

### Complexity of the algorithm

The complexity of the algorithm is measured by the execution source. Every algorithm has to be declared as fully functional only if the purpose of the algorithm is achieved within any controlled environment. There is no standard measure to declare the efficiency of any given algorithm. In general, there are a few factors that can be considered as influencing the effective result of the algorithm.

1. **Time complexity**: It is typically expressed using big O notation. Big O notation is a way of describing the asymptotic behaviour of a function. It helps in understanding the increasing running time (RT) of an algorithm with the size of the input (SOI).

**RT ∝ SOI**

1. **Space complexity**: The space complexity of an algorithm is typically expressed using big O notation. The space complexity of an algorithm measures how much memory the algorithm uses.
2. **Constant factor**: The constant factor of an algorithm is the amount of time or space that the algorithm uses, independent of the size of the input.
3. **Amortized analysis**: Amortized analysis is a technique for analyzing the efficiency of an algorithm that takes into account the worst-case and average-case scenarios.

Every algorithm faces difficulties in certain scenarios. Understanding the scenarios with reference to their limitations and assumptions is crucial for effectively handling these difficulties. As there could never be a single solution for any given situation, there will be multiple options for every algorithm. Selecting appropriate techniques to address the available options and progressing with scenarios is the crucial factor.

### Influencing limitations for any algorithm

It is important to understand the limitations of the algorithm from the context of the implementation. Every algorithm is influenced by the following factors.

1. **Non-linear relationships**: If the relationship between the independent and dependent variables is non-linear, linear regression algorithms may not accurately model the data. In such cases, non-linear regression algorithms or other techniques like decision trees or neural networks may be more appropriate.

2. **Outliers**: Outliers are data points that deviate significantly from the overall pattern. They can have a strong influence on regression models, leading to biased predictions. Robust regression techniques that can handle outliers, such as robust regression or support vector regression, may be needed.

3. **Multicollinearity**: Multicollinearity occurs when two or more independent variables in a regression model are highly correlated. This can impact the interpretability of the model and lead to unstable coefficient estimates. Techniques like principal component analysis or ridge regression can help address multicollinearity.

4. **Missing data**: When there are missing values in the dataset, regression algorithms may struggle to provide accurate predictions. Imputation techniques or algorithms specifically designed to handle missing data, like multiple imputation or maximum likelihood estimation, can be employed to deal with this issue.

5. **Heteroscedasticity**: Heteroscedasticity refers to the unequal spread of residuals across the range of independent variables. It violates the assumption of homoscedasticity in regression models and can result in biased standard errors and inaccurate inferences. Methods like weighted least squares or transforming the dependent variable can help mitigate heteroscedasticity.

### Limitation categories

Most of the limitations for any given algorithm would be categorized into three sections.

1. **Theoretical limitations**: Some problems are theoretically impossible to solve by any algorithm. These problems are called undecidable problems. These problems are a class of problems that cannot be solved by any algorithm. One such problem is the halting problem, which involves determining whether a computer program will terminate or run indefinitely. It is proven to be undecidable, meaning that there is no algorithm that can correctly solve it for all possible programs.
2. **Practical limitations**: Even if a problem is theoretically solvable, there may be practical limitations that prevent us from finding an efficient algorithm for solving it. For example, the travelling salesman problem is the problem of finding the shortest route that visits a given set of cities. This problem is NP-hard, meaning that there is no known polynomial-time algorithm for solving it. The specific limitations that apply to a particular algorithm will depend on the problem that it is trying to solve and the environment in which it is being used. There are 5 major points to consider as high-priority limitations for any given algorithm.
3. **Undecidable problems**: There are some problems that cannot be solved by any algorithm, no matter how well-designed.
4. **Intractable problems**: There are some problems that can be solved by algorithms, but the algorithms take an impractical amount of time to run. For example, to understand the black hole and its properties, it is not possible for human life span. But, astronomers, have produced a network of telescopes, Event Horizon Telescope (EHT) to simulate reality with black holes.
5. **Human limitations**: The development and use of algorithms can also be limited by human factors, such as the availability of expertise and the cost of development.
6. **Bias**: Algorithms can be biased, either intentionally or unintentionally. This can lead to unfair or inaccurate results.
7. **Ethical considerations**: The use of algorithms can raise ethical concerns, such as the potential for discrimination or privacy violations. There exist algorithms to clone a human genome. Yet most of the nations have paused their research due to the uncertainty of the risks and benefits that these studies will yield.
8. **Implementation limitations**: Even if an efficient algorithm exists, it may be difficult to implement it efficiently. This is because the algorithm may be complex or require a lot of resources. For example, Shor's algorithm is an efficient algorithm for factoring large numbers. However, it is difficult to implement Shor's algorithm on a real computer because it requires a quantum computer.

### The context for High-dimensional data

In the context of high-dimensional data, it is important to consider the presence of collinearity among the predictor variables. Collinearity occurs when two or more predictor variables are highly correlated with each other. This can lead to unstable coefficient estimates and difficulty interpreting the regression results.

There is a huge problem in fitting the random data into the related network. One way to address collinearity is by using principal component analysis (PCA) to transform the predictor variables into a set of uncorrelated variables.

The preparation of the algorithms is model-centric. These models are the backbone for the performance of the algorithm and the outputs yielded in conjunction with the trained data. The goal of these algorithms is to build the best model for any given dataset. The information age is the outcome of this combination. By definition, information is the processed form of data by these algorithms. This era is transformed into a Knowledge era. The transformation happened due to the visualisation of various dimensions of the same data.

The number of transformations, perspectives or dimensions increases the difficulty in managing the data. When a known number of dimensions is transformed into a known number, it is often referred to as **high dimensional** data. Perspectives of the same data by various experts yield interesting results and unfurling unknown horizons. Thus, the domain experts experiment with the data continuum to predict the outcomes or future values.

Multi-dimensional data with regression algorithms is a powerful tool for analysing complex continuum or datasets. Regression algorithms are used to identify relationships between different variables in the dataset and can be applied to both linear and nonlinear data. This type of analysis provides valuable insights into the relationship between multiple dimensions, allowing businesses to better understand their customer base, predict future trends, or optimize operational processes.

The challenge with high dimensional data is the curation to fit into the trained data model. The real-world data is generated with a set of characteristics, it is easy to classify and process. Whereas the domain experts are viewing the same data from different perspectives, a new dimension evolved from that viewpoint. Hence, the evolution of high dimensions from various subject matter experts.

### High-dimensional data processing techniques

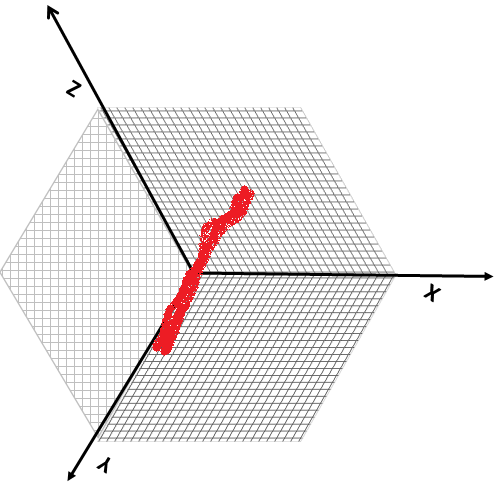
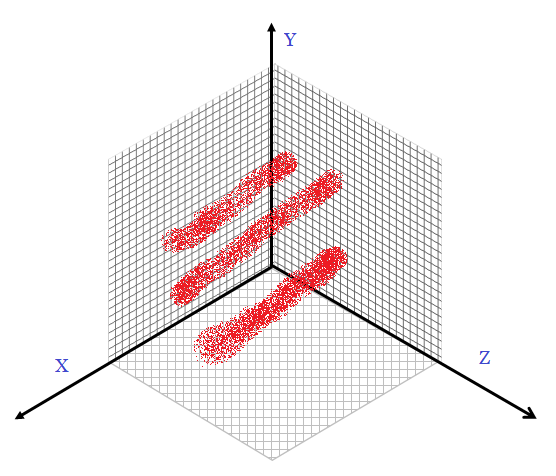
The most common form of multi-dimensional regression is multivariate linear regression (MLR), which uses multiple independent variables as inputs for predicting an outcome variable. MLR models can provide more accurate predictions than single variable models due to their ability to capture interactions among predictors that cannot be detected by other methods such as correlation coefficients or t-tests alone. Additionally, MLR allows researchers or business analysts access deeper levels of insight from large datasets where traditional statistical techniques may not yield fruitful results because they lack sufficient coverage across all relevant factors affecting a given outcome measure.

## Problem statement

The problem with real-world scenarios to solve by these algorithms is twofold. One is identifying the data elements that would be useful, classified as curated data (or) identifying the bad data. The second fold is to derive a better model that fits with most of the properties of the data continuum.

## Situation analysis

Any given dataset with various matrices can be understood (or) interpreted with the help of visual methods by humans, but, not by algorithms. For example, given the data points spread across a 3-dimensional graph would appear as shown in below Picture 2. The same can be viewed when the angle is rotated to some angle on the same data within the same 3-dimensional graph, as shown in Picture 3.



##### Picture 2 Picture 3

##### Picture 2: 3D graph of sample data

##### Picture 3: 3D graph of sample data when tilted to a different angle

Modern machine learning approaches have enabled us to take advantage of even more sophisticated forms of multi-dimensional regressions such as deep neural networks (DNNs) which use layers upon layers of neurons connected to a model with complicated functions that would otherwise require manual feature engineering efforts with conventional methods like logistic regressions etc. DNNs allow us to explore complex patterns within high dimensional space while providing greater accuracy compared to traditional techniques for solving real-world problems related to classification and prediction tasks. Multi-dimensional data with regression algorithms offer immense potential when it comes to understanding our environment better through predictive analytics thereby helping organizations make informed decisions without sacrificing efficiency or accuracy in their operations.

## Core Objective of this study

My main objective of Research in this field is to identify the relation and derive a meaningful outcome from these algorithms when dealing with high-dimensional data. To achieve this goal, it is very much important to understand the initial objectives of these algorithms. Every algorithm is evolved with a purpose and matured over time with many studies and proposals by experts in this field.

Any regression algorithm intends to model the relationship within any given data continuum or a fixed set of data points with various characters exhibited over time. This relation is mainly between a dependent variable (also known as the outcome or response variable) and one or more independent variables (also known as predictors or features) of the given dataset. The goal is to use this model to make predictions with the help of trained data about the dependent variable based on the values of the independent variables.

## Objectives of Regression Algorithms

The main objectives of Regression Algorithms are:

1. To model the relationship between a dependent variable (also known as the response or output variable) and one or more independent variables (also known as predictor or input variables)
2. To make predictions about the value of the dependent variable based on the values of the independent variables
3. To identify the important independent variables that have a strong impact on the dependent variable
4. To identify and account for any *non-linear relationships* or *interactions* among the variables
5. To make *inferences* about the population based on the trained data
6. To identify and handle *outliers* and *missing* data in the dataset
7. To provide interpretable results that can be used for decision-making, conclusions and prediction
8. To understand and estimate the *strength* and *direction* of the relationship between these variables, and how *changes* in the independent variables *affect* the dependent variable
9. To identify *important* independent variables that significantly impact the dependent variable
10. To make *approximately* accurate predictions where the data include missing values or outliers
11. To be able to handle high-dimensional and complex data with large-scale properties or attributes
12. To be able to incorporate domain-specific knowledge and handle categorical variables

Fundamentally, the objectives of regression algorithms are:

###### Estimating the relationship between variables

The model should be able to estimate the relationship between the dependent and independent variables.

###### Making predictions:

The model should be able to make predictions about the dependent variable based on new values of the independent variables.

###### Identifying important predictors and model interpretability:

The model should be able to identify which independent variables are most important in determining the value of the dependent variable. The model should be interpretable and easy to understand so that its predictions can be trusted and its results can be used to inform decision-making

###### Handling various types of data along with missing data:

Regression algorithms should be able to handle various types of data continuum, categorical, and binary data. The model should be able to handle missing data and outliers present in the data.

###### Scalability:

The model should be able to scale to large datasets and work in real-time applications. To understand any algorithm, it must be put into implementation. The algorithm must experiment with variants and try them with real problems. The real-world data continuum will test the scalability.

## Objectives of Regression Algorithms for High-dimensional data

The main objectives of regression algorithms for high-dimensional data are:

###### Handling high-dimensional data by dimensionality reduction:

High-dimensional data can be difficult to visualize and analyze. Dimensionality reduction techniques can be used to reduce the dimensionality of the data while preserving important information. The model should be able to reduce the dimensionality of the data, that is, to identify a subset of the independent variables that are most relevant for predicting the dependent variable.

###### Feature selection with correlated variables:

The model should be able to *select* the most important features among the many available variables.

Regression algorithms for multi-dimensional data should be able to handle correlated variables, which can lead to *multicollinearity* and affect the model's ability to make accurate predictions.

###### Model interpretability & Scalable:

The model should be interpretable and easy to understand, even when dealing with high-dimensional data so that its predictions can be justified and its results can be used to support decision-making. It should be able to scale to large datasets and work in real-time applications.

###### Handling a large number of categorical variables and missing data:

Regression algorithms for multi-dimensional data should be able to handle large datasets which has categorical variables and convert them into numerical variables effectively. The model should be able to handle missing data effectively and not be adversely affected by it. They should have infused with techniques for handling missing data are important for effectively analyzing the data.

###### Understanding the underlying structure of the data by identifying the important features:

Multi-dimensional data often have complex relationships and patterns that are difficult to distinguish in lower-dimensional representations. Analyzing multi-dimensional data will unfurl hidden structures and patterns. While it is often evident it contains a large number of features, it can be difficult to determine which features are most *important* for understanding the relationships in the data.

###### Clustering:

Multi-dimensional data can be used to find groups of similar observations (clusters) in the data.

###### Anomaly detection:

Multi-dimensional data can be used to detect unusual observations that do not conform to the general patterns in the data. Quantum algorithms play a better role in all such detection.

###### Forecasting and prediction:

Multi-dimensional data can be used to create forecasting models or predictions about future events.

## Objectives of Review of Literature

The objectives for reviewing the literature are very important to highlight the significance of the research. The foundation stone of the study is evolving during the period by the researchers at the parallel time. I’m going to review these objectives in the following sections

1. Identifying the strengths and weaknesses of the studies by fellow researchers
2. Comparison of algorithms
3. Identify the best practices and
4. Highlight the trends & future directions

#### Error & Biased Estimators

The limitations of every statistical approach start with a parametric model and develop optimal methods under that model using maximum-likelihood theory. While establishing the maximum likelihood, the error factor must be evaluated. Its evaluation is easy for smaller sets of data, whereas the data sampling sets increase, the complexity of calculation increases. That’s the situation biased estimators evolve and play a role. Every model’s error term is distributed with a mean of zero and a constant variance. This allows for the use of maximum likelihood estimation, to make inferences about the parameters of the model. **Karl F. Gauss** highlighted this concept with practical application in his theory of normal distribution to the measurement of errors. And later became instrumental in other research to lead with this proposition. The error between the average prediction model and the actual growth of the model is considered as **Bias**.

*bias*(x) = 𝔼 [ *f* '(x)] - *f*(x)

The assumption of Gaussian errors is common but the methods may not perform well if the data does not follow exactly the model. This can happen in two scenarios - where the data is not normal but follows a symmetric distribution with heavier tails or where most observed data follow the Gaussian model but a few observations do not (outliers). The field of robust statistics aims to find a good balance between efficiency at the target model and valid inference under model violations by fitting the majority of the data well while limiting the influence of atypical observations.

*Weights & influence in neural networks*

Weight refers to an amount assigned to every item in the sequence to increase or decrease its importance in the given dataset. This concept is crucial for decision-making processes in various functional domains. A higher weight will have a more significant impact on the overall performance of the data. Analysts use this information to make informed decisions about their strategies and optimize the outcomes of their algorithms.

As the data becomes complex and multi-dimensional, the algorithm will face the challenge of dealing with data that is imbalanced without a pattern. This is very usual and happens when one set of data is significantly more prevalent than another. For example, in a financial trading dataset, the number of buyers may far outnumber the number of sellers with a particular influence on the market

Imbalanced data will always be a problem for any algorithm, as they tend to be biased towards the majority class. This can result in poor performance when it comes to predicting the minority class. However, there is a solution to this problem: using weights. Weights can be used to balance imbalanced data by giving more importance to the minority class and less importance to the majority class. This means that the algorithm will pay more attention to the minority class, resulting in better predictions for that class.

To implement weights, one can assign a higher weight to the minority class and a lower weight to the majority class. This can be done in various ways, such as using the inverse of the class frequency or using a more complex algorithm such as SMOTE (Synthetic Minority Over-sampling Technique). Such techniques can improve the performance of your algorithms when dealing with imbalanced data. This can be particularly useful in industries such as healthcare, finance, and fraud detection, where the minority class is often of greater importance.

While working with linear regression (or) logistic regression, these weights represent the coefficients of the input features towards generating the outcome. The linear regression model with a series of input features (X1, X2, … Xn) can be represented as

y = w1 X1 + w2 X2 + ... + wn Xn + b

where w1,w2 .. wn are the weights associated with each input feature and b is the bias term.

In healthcare, statistical weight is used to analyze patient data and identify disease risk factors. By assigning different weights to different variables, healthcare professionals can determine which factors are most likely to contribute to a patient's health outcomes.

Scholars in the healthcare industry should use this information to develop more effective treatment plans and improve patient outcomes. By understanding the significance of statistical weight, algorithms can make informed decisions that optimize their performance and improve their outcomes.

While constructing neural networks in the field of artificial intelligence, weights play a crucial role. Weights are essentially numerical values that determine the strength of the connection between nodes in a neural network. These connections allow information to flow through the network and ultimately produce an outcome of the algorithm. The significance of weights lies in their ability to influence the outcome of the network. The higher the weight, the more influence an input will have on the output. Conversely, a lower weight will result in less influence. By tweaking the weights, algorithms can fine-tune the network to produce more accurate results.

## Co-Relating the objectives of research to the objectives of the Review of the Literature

A typical algorithm usually accepts or denies the data element depending on the selected feature and proceeds further without deviating due to the inflow from the data continuum. Whereas, for a high dimensional data continuum, the algorithm has to find a suitable switch when the relative features of altering dimensions. To make it simple, if we consider

**dataset-1** as {1,2,3,4,5,6,.....n} => ∀ n used { n ∈ R ∣ n > 0 } or { n ∈ R: n > 0 } is the set of all n ∈ R where n > 0 (n is greater than zero 0)

*f*(d1) = { n: *n*i=*n*(i-1)+1 ∀ i>0, n ∈ R and n>0}

**dataset-2** as {1.03,0.85,1.08,0.62,1.05,1.17,1.11,1.15.....x} => ∀ x as a rational number, x used { x ∈ ℤ ∣ 0>x<2} or { x ∈ ℤ: 0>x<2} is the set of all x ∈ ℤ where x>0 (x is greater than zero 0)

*f*(d2) = { x: where m,n ∈ ℤ ∀ m,n ≠ 0, x ∈ R and 0>x>2}

If the algorithm has to work on these two different datasets, at the feed of every data point, the membership of the data element is changed along with the properties. Thus there is no unique feature that can be considered as the primary focal point. Hence, the algorithm has to switch to the analysis of the data continuum according to the input inflow.

During the inflow, at times, there could be bad input data. Identifying that bad element is the key point of the analysis by the algorithm. When the machines are used to make predictions with the help of algorithms, it is important to make many choices about *how the machine works*. These choices can affect *how well the machine works*. So, it is vital to try different choices and find the best ones to make the machine work the best it can. A practical study by Ari, Lukumon etc., [[Ref.1](#Ref1)] that compared the performance of rainfall forecasting models based on LSTM-Networks architectures with modern algorithms used historical weather data from five major UK cities and implemented good practices such as pre-processing procedures, feature selection, and hyperparameter grid search to obtain the best performance of the models. The results showed that the Bidirectional-LSTM and Stacked-LSTM models achieved lower values in the evaluation metrics and can be used as rainfall prediction models, but the models still struggle to adapt to abrupt variations in precipitation patterns.

Within the unsupervised learning algorithms, a self-organizing map (SOM) is used to identify the patterns and structure in high-dimensional data. This algorithm generates a low-dimensional representation of the input data, typically in the form of a two-dimensional grid of nodes or *neurons*, which are arranged in a way that preserves the topological structure of the data. The initial grid configuration, neighbourhood size, choice of learning rate and other hyperparameters limit the performance of this algorithm. This algorithm functions well when the dimensional reduction is achieved with the rate of data infusion.

The SOM-based algorithm was able to outperform a traditional regression algorithm, and it was also more robust to noise and outliers. In a study published in the journal "Neurocomputing", SOMs were used to reduce the dimensionality of a dataset of medical images. This allowed to train algorithm to predict better ever.

The static architecture and limited capabilities for representing hierarchical relations in high-dimensional data by self-organizing map (SOM) didn’t stop the application analysts from experimenting with data mining techniques. In 2002, Andreas, Dieter and Michael developed a layered architecture proposal as Growing Hierarchical SOM (GHSOM) [[Ref.2](#Ref2)] that addresses both of these limitations. The GHSOM is an artificial neural network model with a hierarchical architecture composed of independent growing SOMs. This model adapts its architecture during its unsupervised training process to meet the specific requirements of the input data. Additionally, the global orientation of the independently growing maps in the individual layers of the hierarchy facilitates navigation across branches. This is a problem-dependent architecture type of neural network representation of hierarchical relations in the data. This is especially appealing in explorative data mining applications, allowing the data’s inherent structure to unfold in a highly intuitive fashion.

The relations within data elements of any given dataset are identified with a feature or a dimension. In a data continuum, it is difficult to identify and utilize which feature of the dataset would be useful by the algorithm towards yielding the outcome. In 2014, Josua, Adam and Enrico published a study, INFUSE, [[Ref.3](#Ref3)] a novel visual analytics system that ranks the predictive features across feature selection algorithms, cross-validation folds and classifiers. It is designed to be domain-independent for that time, yet the performance is influenced when the number of cross-validation folds is more than ten. A logical analysis of these data elements with reference to their features will help in improving the structure and performance of the respective algorithms. This analysis plays a vital role in switching the execution route. While the algorithm is executing the logic and the data is continuously influencing the analysis with its variety of features, the algorithm has to align with inflow.

A computational algorithm uses calculations to derive and output new data from another set of input data.

Completed

## Review Related Studies

## Research Gap

There are several areas where further research on regression algorithms is needed:

1. Handling of missing data: Current methods for handling missing data in regression models have limitations, and there is a need for more effective approaches.
2. Handling of high-dimensional data: With the increasing amount of data being collected, there is a need for regression algorithms that can effectively handle high-dimensional data and large numbers of features.
3. Handling of non-linear relationships: Many real-world problems involve non-linear relationships between variables, and there is a need for regression algorithms that can accurately model these relationships.
4. Handling of categorical variables: Current methods for handling categorical variables in regression models can be limited, and there is a need for more effective approaches.
5. Handling of time series data: Time series data can be difficult to model using traditional regression algorithms, and there is a need for more specialized methods for handling time series data.
6. Scalability: With the increasing size of datasets, there is a need for regression algorithms that can scale to handle large datasets.
7. Explainability: As Machine Learning models are widely used in decision-making, There is a need for regression algorithms that can provide insights into the predictions and decision-making process.
8. Handling high-dimensional and complex data: Traditional regression methods may not be able to handle large amounts of data or data with complex relationships.
9. Incorporating prior knowledge: There is a need for methods that can incorporate domain-specific knowledge into the regression process.
10. Handling missing data: Traditional methods may not be able to handle missing data effectively.
11. Scalability: There is a need for methods that can scale to large datasets and work in real-time applications.
12. Handling non-linear relationships: Traditional linear regression methods may not be able to capture non-linear relationships in the data.
13. Handling categorical variables: Traditional methods may not be able to handle categorical variables effectively.
14. Robustness: There is a need for methods that can handle outliers and noise in the data.
15. Explainability: There is a need for methods that can provide interpretable and explainable results.

## References

Ref.01 : Barrera-Animas, Ari & Oyedele, Lukumon & Bilal, Muhammad & Akinosho, Taofeek & Davila Delgado, Manuel & Akanbi, Lukman. (2021). Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. Machine Learning with Applications. 7. 100204. 10.1016/j.mlwa.2021.100204.

Ref.02 : Rauber, A., Merkl, D., & Dittenbach, M. (2002). The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. IEEE Transactions on Neural Networks, 13(6), 1331–1341. doi:10.1109/tnn.2002.804221

Ref.03 : Krause, J., Perer, A., & Bertini, E. (2014). INFUSE: Interactive Feature Selection for Predictive Modeling of High-Dimensional Data. IEEE Transactions on Visualization and Computer Graphics, 20(12), 1614–1623. doi:10.1109/tvcg.2014.2346482

Ref.04 :

Ref.05 :

Ref.06 :

Ref.07 :

Ref.08 :

Ref.09 :

Ref.10 :

Ref.11 :

Ref.12 :

Ref.13 :

Ref.14 :

Ref.15 :

Ref.16 :

Ref.17 :

Ref.18 :

Ref.19 :

Ref.20 :

Ref.21 :

Ref.22 :

Ref.23 :

Ref.24 :

Ref.25 :

Ref.26 :

Ref.27 :

Ref.28 :

Ref.29 :

Ref.30 :

Ref.31 :

Ref.32 :

Ref.33 :

Ref.34 :

Ref.35 :

Ref.36 :

Ref.37 :

Ref.38 :

Ref.39 :

Ref.40 :

Ref.51 :

Ref.52 :

Ref.53 :

Ref.54 :

Ref.55 :

Ref.56 :

Ref.57 :

Ref.58 :

Ref.59 :

Ref.60 :

Ref.61 :

Ref.62 :

Ref.63 :

Ref.64 :

Ref.65 :

Ref.66 :

Ref.67 :

Ref.68 :

Ref.69 :

Ref.70 :

Ref.71 :

Ref.72 :

Ref.73 :

Ref.74 :

Ref.75 :

Ref.76 :

Ref.77 :

Ref.78 :

Ref.79 :

Ref.80 :

Ref.81 :

Ref.82 :

Ref.83 :

Ref.84 :

Ref.85 :

Ref.86 :

Ref.87 :

Ref.88 :

Ref.89 :

Ref.90 :