Coventry University

Modelling and Optimization Under Uncertainty

7135CEM – Coursework

Jordan Bignell & Georgios Chantziplakis
22/04/2021

Contents

Comparing the behaviours of Two Topic Modelling Algorithms on COVID-19 Vaccination Tweets: LDA vs LSA

Georgios Chantziplakis
Faculty of Engineering, Environment and Computing
MSc. Data Science and Computational Intelligence
Coventry University, Coventry, United Kingdom
Chantziplg@uni.coventry.ac.uk

Abstract – Coronavirus is a newly developed infectious disease that has consequently caused a pandemic due to its ease of transmission as of early 2020. Several companies and organizations from around the world have been working countlessly to produce a vaccine to prevent and avoid the spread of the virus. This article contains Tweets from Twitter regarding the opinion about the vaccine. This paper explains and shows the topic modelling techniques to categorize the tweet by subject and gain a better understanding of the vaccine through a positive or negative sense. Latent Dirichlet Allocation and Latent Semantic Analysis were implemented as Topic Modelling techniques and concluded that Latent Dirichlet Allocation was the superior unsupervised technique for categorizing the text in 12 topics.

Keywords – COVID, Latent Dirichlet Allocation, Latent Semantic Analysis, Unsupervised Machine learning, Twitter

Introduction

As of early 2020, Coronavirus is a newly formed infectious disease and due to the ease of transmission has caused a pandemic. The official medical name for Coronavirus is COVD-19, which will be referred to throughout this paper (World Health Organisation, 2020).

This pandemic has forced the majority of the world to change and adapt to a new lifestyle to avoid the spread of infection of this newly formed disease. It is found that older people and those with underlying health conditions such as diabetes and cardiovascular disease have a higher chance to develop more impactful symptoms that could be life-threatening. (World Health Organisation, 2020).

Throughout this epidemic, many organizations from different countries have been developing a vaccine that can prevent and stop the spread of this virus. This would allow the public to return to normality, however, the vaccine cannot be developed and produced instantly due to the testing and rigorous experiments to ensure the safety of the vaccine. Due to the fierce and rapid development of the vaccine, many people have concerns. Therefore, obtaining the tweets to decipher and categorize each topic will provide us with an understanding of the views, and opinions of the vaccine, if this is positive or negative feedback.

Topic modelling is an unsupervised machine learning approach used for detecting abstract topics contained in a collection of text documents. The two main topic models used are Latent Dirichlet Allocation (LDA) and Latent Semantic Analysis (LSA). These techniques can detect words and phrase patterns, scanning documents, and automatically clustering word groups and similar expressions (Pascual, 2019).

This paper aims to study and investigate the behaviour of LDA and LSA, on COVID-19 Tweets which were published between December 2020 and April 2021, and in addition, understanding the topics categorized by the tweets. We expect to learn the subject areas from each vaccine that has been included in this dataset. Furthermore, taking into consideration the social, ethical, and legal context of COVID-19.

Literature Review

In this publication by Garcia & Berton (2021), the paper was based on tweets in both English and Portuguese language, which from research studies have only been conducted in one language. The modelling techniques used in this paper were Latent Dirichlet Allocation (LDA) and Gibbs Sampling Dirichlet Multinomial Mixture (GSDMM) which is better suited for short text due to the lack of word co-occurrences.

For Sentiment Analysis specifically, CrystalFeel and SBERT are used to capture the range of emotions used in the text such as joy, sadness, anger, or fear.

Due to the language difference, it is worth noting the change of keywords of certain subject areas such as 'pandemic' in English and 'Pandemia' in Portuguese, and 'Quarantine' and 'Quarentena' respectively. Therefore, teaching the system to understand these terms is important to learn and correctly categorise each topic. Alternatively, creating 2 separate systems to work dependable on language could provide better results. In total there was a recorded

7,144,349 English tweets and 7,125,530 Portuguese tweets, furthermore, conducting the machine learning techniques on language could output more defined results given the large dataset for each language, respectively.
From the results section of the paper (Table 6 & 7, Garcia, K., & Berton, L. 2021), it is clear from the 10 selected topics of Economic impact, Case reports, Proliferation care, politics, entertainment, treatments online events, charity, sports, and anti-racism protests, that the Portuguese topics have been categorized using more words in comparison to the English topics for example:
English topics for Economic impact: Work, impact, business, crisis, pay.
Portuguese topics for Economic impact: Buying, money, working, crisis.
In addition to this, the reasoning could be because of the definition of the term, if we look closer at the terms for example 'Buying' (Portuguese topic) and 'Pay' (English topic) that both of the terms are for purchasing however they are picked up in different languages.
When we look at the sentiment analysis of emotions, we find that in the paper, that the majority of tweets were classified as fear for each topic, it is understandable for certain categories such as treatments, as the public does not know what the side effects could be or when they may receive the treatment.
Overall, we find that the topic modelling was successful due to the high rate of topics detected, accuracy and precision however as previously discussed if the model is split by language the results would improve. (Garcia & Berton, 2021)

It is important to understand the problems of using an unsupervised machine learning approach to gain learnings before approaching the main problem, and this is explored using a similar dataset by Haupt et al. in the paper called Characterizing Twitter user topics and communication network dynamics of the "Liberate" movement during COVID-19 using unsupervised machine learning and social network analysis (Haupt et al., 2021).
The unsupervised approach undertaken in this paper was Biterm Topic Model (BTM) where it is again specialized to suit short text data, this technique is useful for the 280-character limit set by Twitter.
From the result of using BTM, can be argued that the topics it has chosen can be defined further. Additionally, topics 2, 3, 4, 5, and 9 are all topics called 'Backlash against Trump Administration', therefore all these topics can be condensed into one as opposed to multiple categories. Therefore, altering and adjusting the topics to be able to receive more refined topics would be more appropriate for this model. Furthermore, concerning our project, it is important that we understand and investigate the topics once the system has processed the tweets due to the unsupervised nature of this technique, the machine may need additional support to be able to identify the topics.

Project Setup & Environment

We work using Python 3.8 on Jupyter Notebook, the main libraries used are Gensim, Sci-Kit Learn, Pandas, NLTK, and PyLDAvis.

Dataset & Problem

The dataset chosen is a real-life problem given the current climate of the world from the pandemic. Understanding, analyzing, and gaining additional insights of COVID-19 would expand our horizons of the new virus which is still relatively new to many of us.
The dataset is from Kaggle an open-source website: Link to dataset (Preda, 2021).

1) Exploration

Firstly, we investigate the data to gain further insights, this is done by understanding the shape of the dataset which includes 65088 records and 16 features. Followed by using descriptive analysis on the data that counts, finds the average mean, minimum and maximum data points. This can identify anomalies within the dataset in certain attributes and how much of an impact this can have. For example, a user with 1,000 followers in comparison to someone with 10 followers is likely to have more of an impact from their 'tweet' because their audience is larger. However, with this problem, we are analyzing the topics of text from the 'tweet', the feature of this column is called 'text'. Each tweet contains up to 280 characters which can contain numbers, string, and symbols. It is well-known of Twitter to use hashtags (#) to join a communication globally or have your opinion heard from a community of similar-minded users. From the shape of our data, we understand that in at least 1 of the columns there are over 65,000 records and we understand if there are any null values in any of the columns, however, most importantly if there are any in the column 'text'.

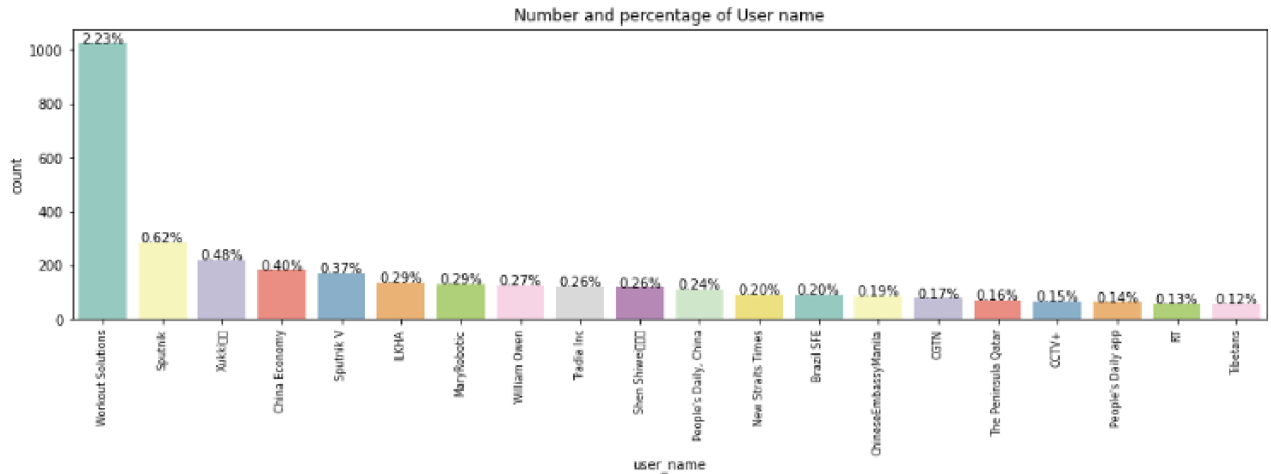


Figure 1 - Number & percent of user's tweet

2) Feature Selection & Pre-processing

As the project is based on topic modelling, we plan to only use the 'text' feature that is based on tweets for each user. To pre-process the text, we firstly format our text and remove stop words in using the English Language. Additionally, we added further stop words that were not included in the original library of StopWords, and this can be showcased in the appendix. Dependable on other languages we would change the language preference on stop words to a different language. A stop word is a commonly used term in a sentence such as the word 'for' or 'the'.

In our dataset, each tweet included a link to Twitter to the desired tweet and this was shown like 'https://t.co/xeHhIMg1kF'. Therefore, this could affect the classification of topics and was required to be removed. The deletion of mentioned users in the tweet was removed for example '@BorisJohnson'. Further other cleaning techniques included removing the hashtag symbol, other symbols, double spaces, Unicode, punctuation, and converting tweet into lowercase. We use lemmatization to keep specific words if they are adjectives, nouns, verbs, or adverbs, after they have been lemmatized. For example, all the words that contain 'shipped' and 'shipping' would be converted to 'ship'.

Doing this pre-processing allows the machine to interpret the text easier in one format. Ensuring all the text has no special characters and is all in lowercase enables the machine to learn easier with less computational complexity.

3) Sentiment Analysis

Sentiment analysis is a technique for determining whether data is positive, negative, or neutral using natural language processing. It is often used on textual data to assist companies in tracking brand and product sentiment in consumer reviews and better understanding the customer needs. Natural language processing (NLP) and machine learning algorithms are used in sentiment analysis to automatically assess the emotional tone of online conversations. In our case, will give us a better insight into the data we are dealing with. (MonkeyLearn, n.d.)

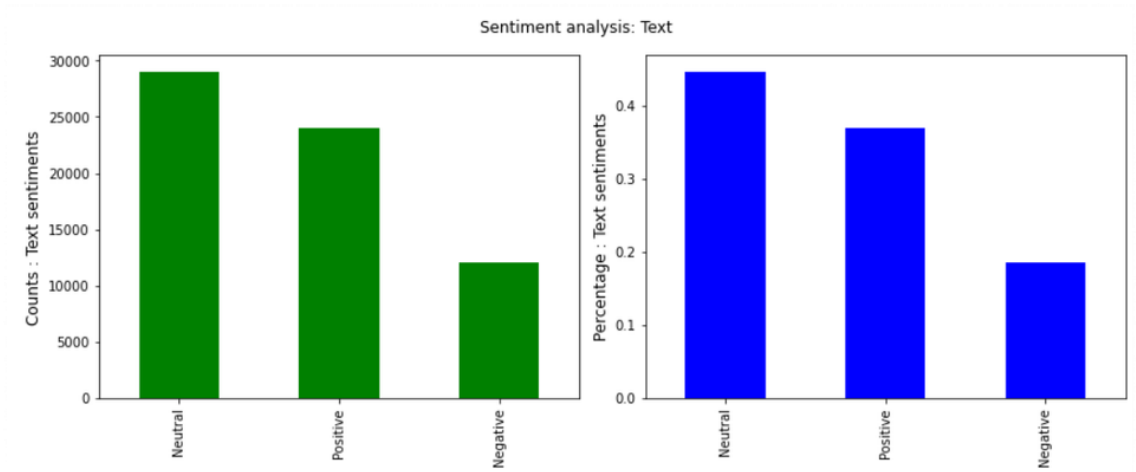
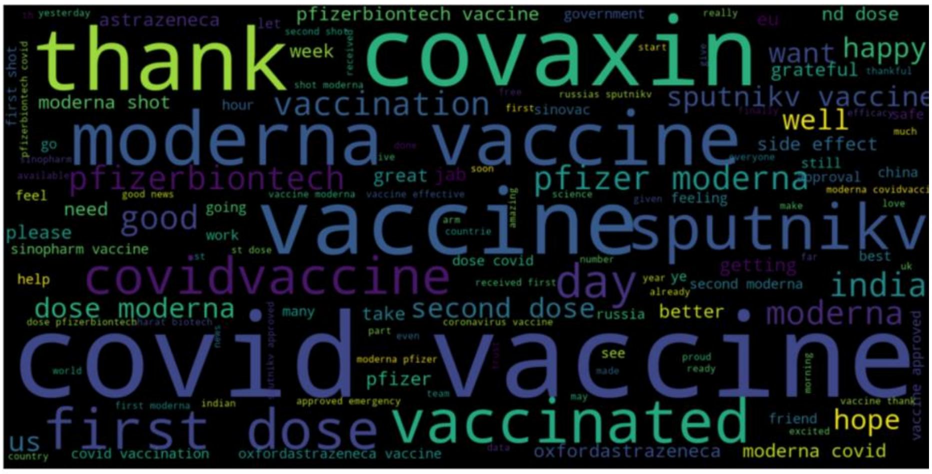


Figure 2 - Sentiment analysis (Neural, Positive & Negative) – Count (Left) & Percentage (Right)



Most Frequent words in tweets (Positive sentiment)

Figure 3 – Prevalent words of Positive Sentiment

Topic Modelling

Topic modelling is a machine learning technique that analyses text data automatically to identify cluster terms for a series of documents. Since it does not include a predefined set of tags or training data that has been previously identified by humans, this is referred to as "unsupervised" machine learning. To infer topics from unstructured data, topic modelling involves counting words and grouping related word patterns. A topic model clusters input that is similar, as well as terms and phrases that occur most frequently, by detecting patterns such as word frequency and distance between words.

The way Topic Modelling works is by dividing a corpus of documents in two parts:

- The topics covered by the documents in the corpus.
- The several collections of documents from the corpus, organised by the subjects they address.

(Pascual, 2019)

To evaluate topic modelling techniques, from research it is found that coherence score is the best method to measure the performance.

The Coherence score is a measurement to indicate the optimal model, this is done by 2 measuring components: Intrinsic and extrinsic measure.

Intrinsic measure is showcased as UMass, it measures to contrast a word only to the preceding and succeeding word, respectively.

Extrinsic measure is represented as UCI. Here all the individual words are paired with every other single word.

Using both the intrinsic and extrinsic measure, it calculates and computes the coherence score, summing the pairwise scores on the words (word 1…Word n). (Kumar, 2018)

In summary, it is unlikely to get a coherence score of 1 or 0.9 unless the words are identical or bigrams e.g., United and States. Below is a loosely summarized representation of the coherence score:

0.3 - Bad
0.4 - Low
0.55 - Okay
0.65 - Good
0.7 - Ideal
0.8 - Unlikely
0.9 - Probably wrong

(Sara, 2019)

Methods

The methods we intend on implementing to our model are Latent Dirichlet Allocation, Latent Sentiment Analysis, and Hierarchical Dirichlet Process. The use of Hierarchical Dirichlet Process is implemented, and we acknowledge that this is a non-parametric Bayesian technique, however we wanted to understand number of topics given from the machine. To evaluate and measure our models, we use coherence score and t-SNE to visualise how accurate and precise the unsupervised techniques are.

1) Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is used to classify text in a document to a specific topic. It creates a Dirichlet distribution-based topic per document and word per topic model. The concept is based on the fact that records are made up of a random mix of latent topics. (towards science). The Dirichlet distribution is a type of probability distribution. Dir () is a class of continuous multivariate probability distributions with a vector of positive reals as a parameter. The Beta distribution is a multivariate generalization of it. In Bayesian statistics, Dirichlet distributions are often used as prior distributions. LDA is an unsupervised machine-learning model that also tells in what percentage each document discusses each subject by giving the wights of the most frequent words of each topic. This is the probability of a word belonging to a specific topic and finally, the 'title' of each topic can be decided based on the most frequent words given by the model.

One of the main advantages of LDA is that when the model has completed its run, it is ready to assign topics to any text. One of the main drawbacks of LDA is that needs human interpretation. By that we mean that the model provides the splitting of the topics but in the end a human need to label them in order to present the results to non-expert's people.

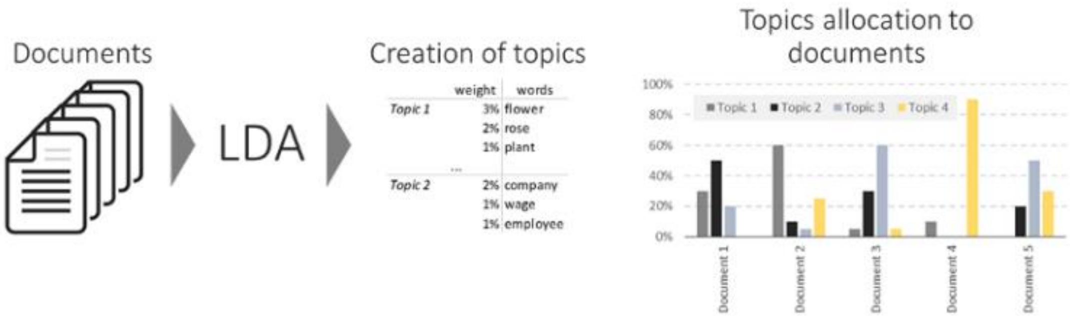


Figure 4 - LDA Example (Revert, 2019)

2) Latent Semantic Analysis (LSA)

One of the fundamental techniques in topic modeling is latent semantic analysis (LSA). The basic concept is to decompose a matrix of what we have — documents and words — into two separate document-topic and topic-term matrices. Creating our document-term matrix is the first step. We can create a mxn matrix A with m documents and n-words in our vocabulary, where each row represents a document and each column represents a phrase, given m documents and n-words in our vocabulary. Each entry in the most basic version of LSA can simply be a raw count of how many times the j-th word appeared in the i-th text. LSA models usually use a tf-idf score instead of raw counts in the document-term matrix. Tf-idf (word frequency-inverse document frequency) assigns the following weight to term j in document I as follows:

$$tf\,idf\,(t, d, D) = tf\,(t, d)\ .\ idf\,(t, D)$$

Figure 5 – TF-IDF Formula (Stecanella, 2019)

Intuitively, a word has a lot of weight if it appears a lot in the text but not so much in the corpus. We want to reduce the dimensionality of our document-term matrix to identify the few latent topics that capture the relationships between the terms and documents. Truncated SVD can be used to perform this dimensionality reduction. Singular value decomposition (SVD) is a linear algebra technique that factors any matrix M into the product of three matrices: M=U*S*V, where S is a diagonal matrix of M's singular values. (Xu, 2018)

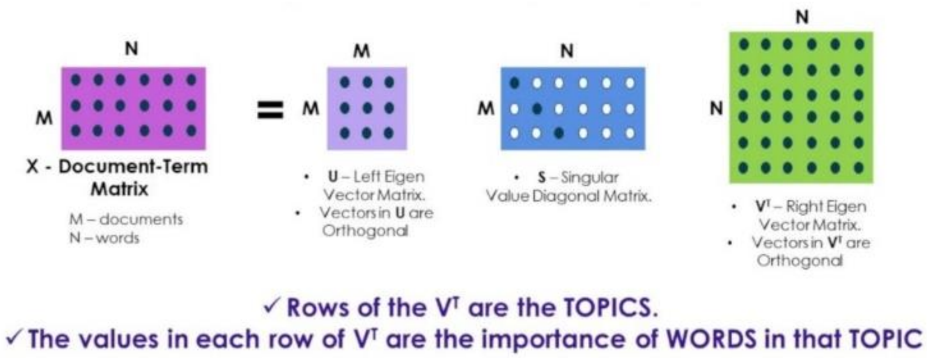


Figure 6 - LSA Example (Arvindpdmn, 2020)

3) Hierarchical Dirichlet Process (HDP)

Hierarchical Dirichlet Process (HDP) is an extension of LDA and where the machine decides on the optimal number of topics from the text data.

The Dirichlet process is used to capture the undefined number of topics. The advantage of using HDP is that the maximum number of topics can be unbounded and learnt from the data, as opposed to being specified before the running of the model. Due to the unsupervised nature of this model, it is more complicated to implement and not necessary in the case where a bounded number of topics are acceptable. (Goodman, 2014)

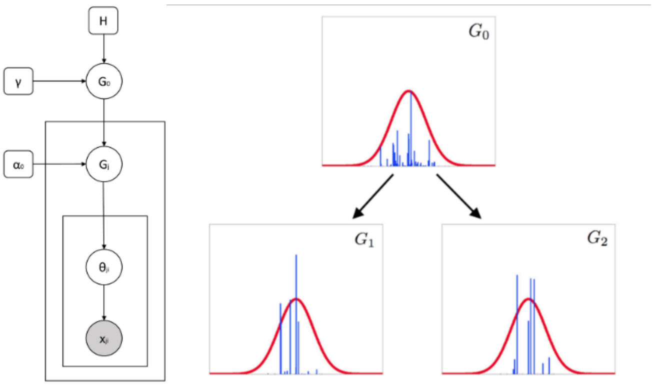


Figure 7 - HDP Example (Sroka, 2020)

Results

From exploring, pre-processing, and analyzing the data, here we showcase the results from our models.

The coherence score is measured against the number of topics, to identify the optimal number of topics for each model, we plot the graph that shows the coherence score by number of topics, this can be evidenced in figure 8 & 9.

In figure 11 & 12, we compare the T-SNE of LSA and LDA of 12 number of topics, to understand the comparison and how well the topics have been accurately modelled.

Figure 10, is an interactive figure that shows the number of words that appear in a certain topic. As shown, we see highlighted in red the words contains in topic 1.

Table 1 showcases the exact coherence score for each model by 4 decimal points.

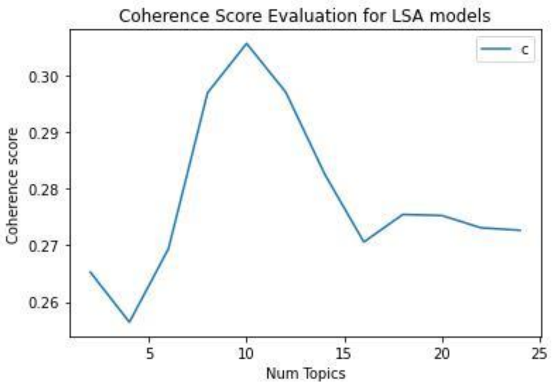


Figure 8 – LSA number of topics / Coherence Score

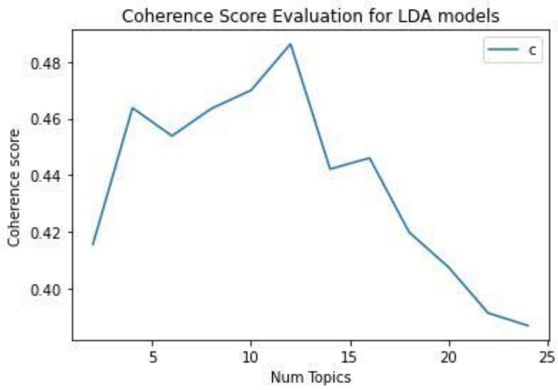


Figure 9 – LDA number of topics / Coherence Score

Table 1 – LDA & LSA results

Number of Topics	LDA	LSA
4	0.4637	0.2564
6	0.4538	0.2694
8	0.4635	0.2964
10	0.4699	0.3038
12	0.4863	0.3015
20	0.4073	0.2812

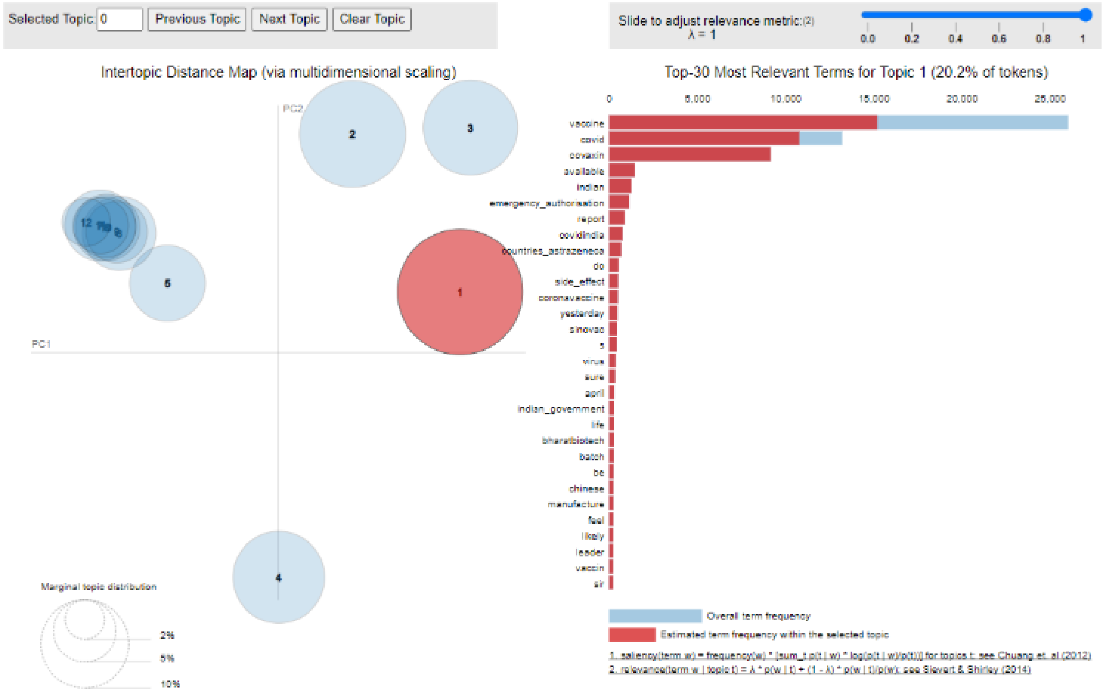


Figure 10 – Number of topics and the most common words

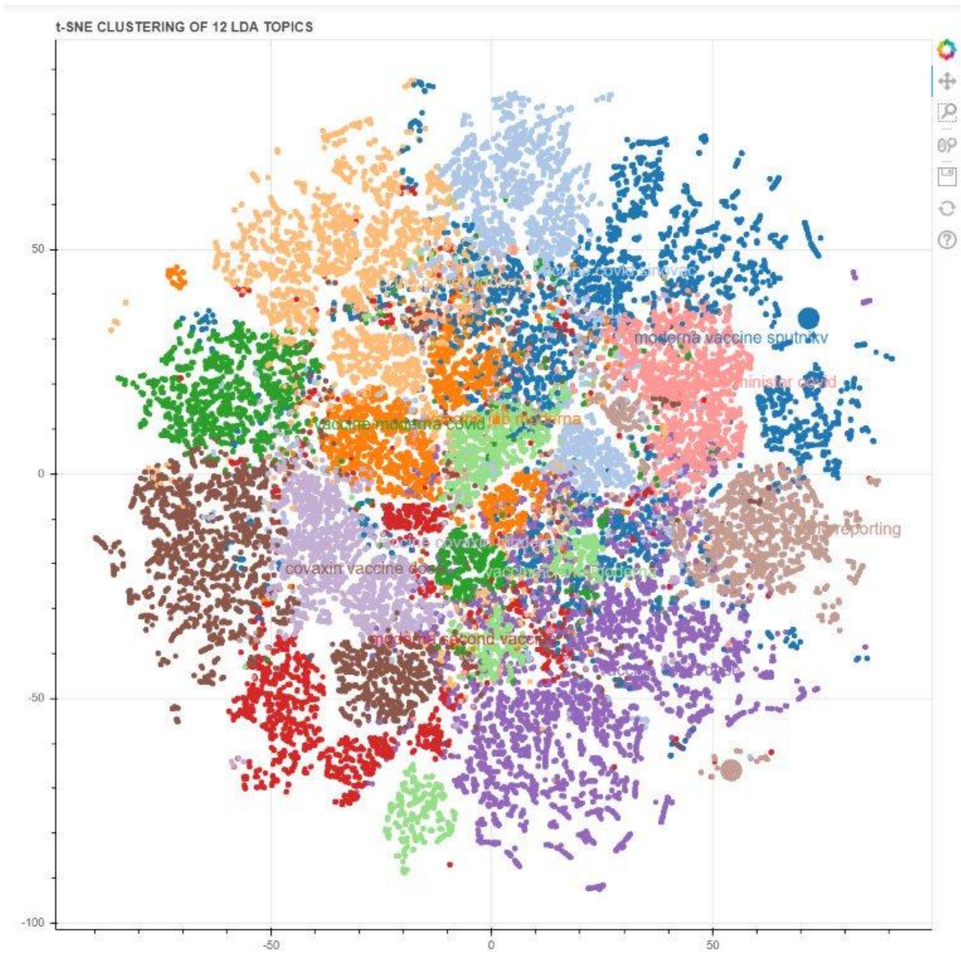


Figure 11 – Cluster of topics from LDA model using the best model (12 topics)

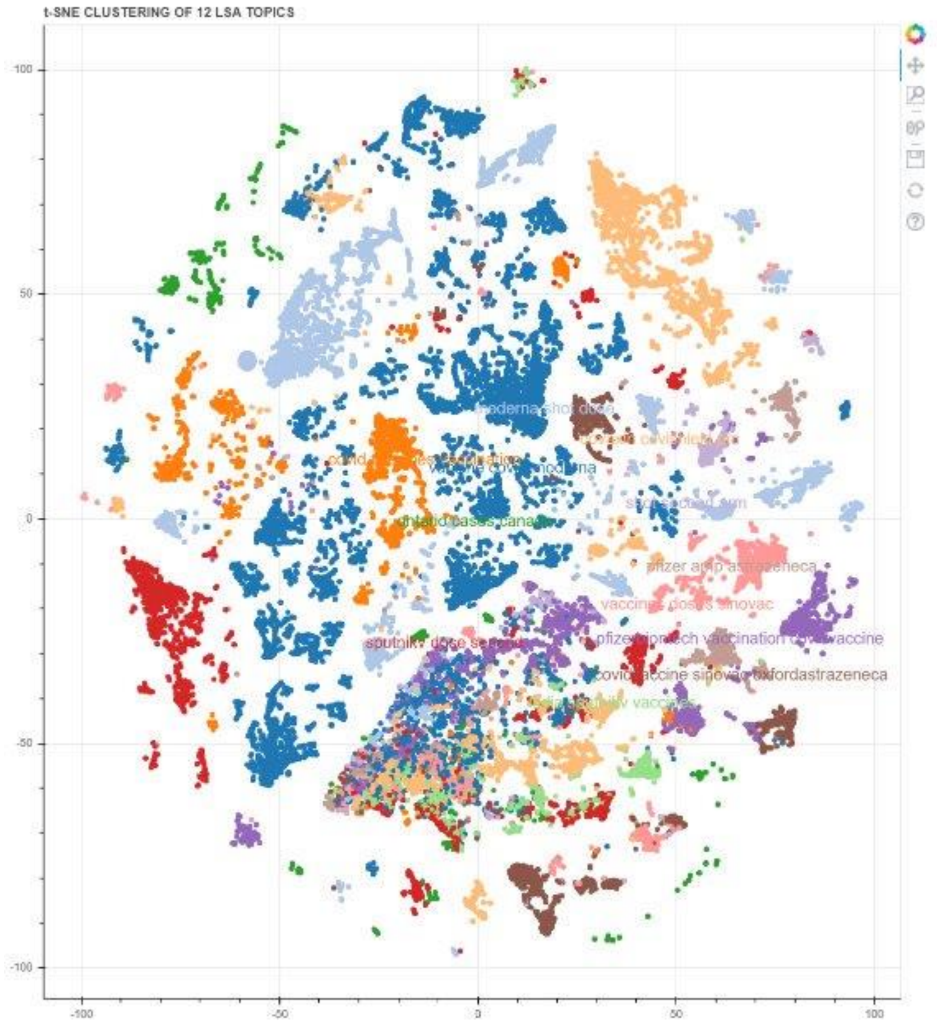


Figure 12– Cluster of topics from LSA model using the best model (12 topics)

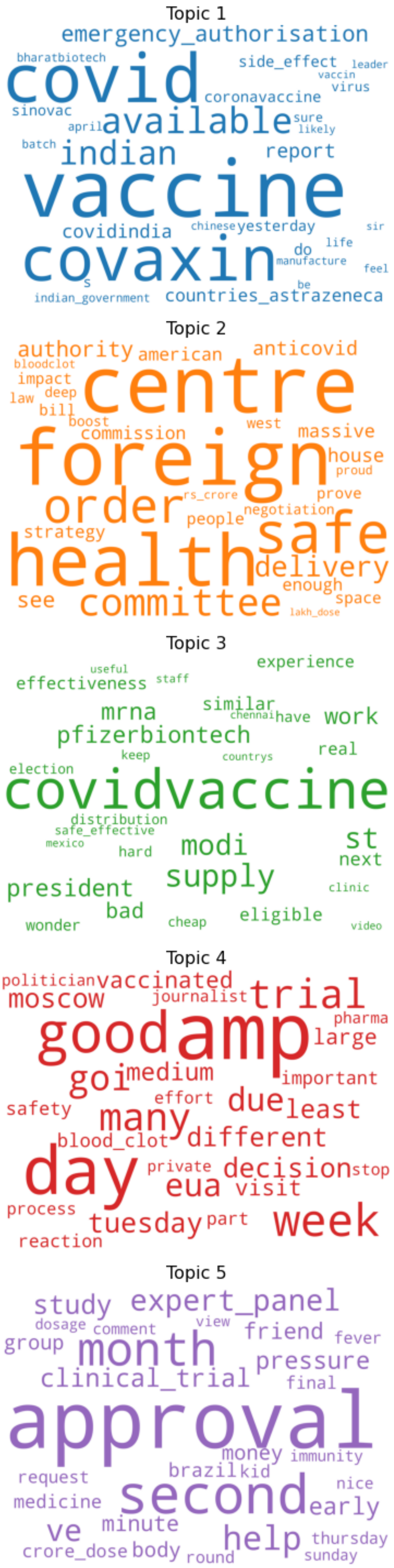


Figure 13 - Most common words for the first 5 topics for LDA

Discussion

Our expectations at the beginning were for our model to categorize each topic by vaccine, for example, Pfizer, AstraZeneca… However, the topics using LDA are similar with some topics being categorized by what looks like country of conversation such as if the tweet is regarding a Russian vaccine, it is categorized in that particular topic. In addition, we did not expect to obtain great results due to the broad and wide subject in which COVID-19 is. Further, our aim was to understand the sub-topics within this large area however, we understood that it would be more challenging for our model to adapt and learn the sub-categories of COVID-19.

Using LSA and LDA we need to find the optimal number of topics as opposed to the HDP approach where the model decides the 'best' number of topics to use. These unsupervised approaches do require some assistance with trial and error to experiment and identify the 'best' number of topics to classify the text.

In figures 8 & 9, we find the optimal number of topics by coherence score for LSA is 10 and for LDA is 12. However, from the summarized representation our LSA score of 0.3 is a bad score and will not categorize the text correctly therefore the LDA approach of 0.48 is a more improved model and provides better accuracy and precise topics that are relevant to each other. Comparing figure 11& 12 (TSNE LSA AND LDA), we find that the LDA is much clearer and shows clarity in terms of understanding each topic in contrast with LSA the T-SNE is messy and unorganized and this links directly to the low coherence score that LSA produced

When HDP is used, it gave us a result of 20 topics which in comparison to LDA, using 20 topics was the worst-performing coherence score. Therefore, allowing the model to decide the number of topics is a bad fit for this approach. As these models are unsupervised, we are required to assist the system to identify the best number of topics.

For both LSA and LDA we compare the coherence score in table 1, and we see the difference that LDA is significantly better than LSA and this could be because LSA concentrates more on reducing the matrix dimensions where LDA focuses and solves the topic modelling issue.

When investigating the social effects and considerations of this topic, due to social media with the new forms and emerging technology and data, the public have more of an opinion. However, the views, social interactions, and communications held over social media means people are able to use their freedom of speech to publicly voice their views on the subject matter, and in this instance, Twitter is the social media platform used. Other concerns can be raised when investigating the ethical, legal, and professional context of COVID-19. When a legal context is portrayed, we look at the regulations and rules of this subject matter and if the tweets are violating any of the code of conducts, which ensures there is no hate speech or other kinds of crime being committed online then there would not be able legal concerns with this.

The use of genism is used to create each model and SKLearn is used to produce the graph T-SNE to represent the coherence score for each number of topics we have trials.

T-SNE abbreviated from T-Distributed Stochastic Neighbor Embedding, is a visualizing tool for high dimensional data. Converting and covering similarity points between data points to join probabilities. Figure 11 can be examined to view our 'best' model and the clarity of the topics being categorized. Other T-SNE's can be evidenced in the appendix.

Ethical, Legal, and Social Issue

Our dataset was collected from Twitter, which is a public networking service with registered users who already have accepted the terms and conditions of the platform. This allows us to collect and subtract information from their tweets without having to worry about any ethical and legal issues.

Conclusion

We believe this project to be original for the fact that there are very few papers that focus on topic modelling for COVID-19 Tweets.

In conclusion, we have discussed thoroughly the techniques of how they work and the application of this to our model. The results show that LDA is clearly a more preferable and stronger method for topic modelling. The efficiency of LDA especially in terms of T-SNE was highly competent. In comparison to the LSA models for all of the topics giving the highest performing of 10 topics at 0.3038 coherence score which is considered low and not suitable for topic modelling. Overall, we identified the robust algorithm of LDA to be superior and more suitable for this project.

Appendix

Link to code and screenshots on One Drive: https://1drv.ms/u/s!Ajl5nOLsDGcBgoYzMuXfSdWpx4SJkA?e=K2JeFl

References

World Health Organisation. (2020). Naming the coronavirus disease (COVID-19) and the virus that causes it. Who.int. Retrieved 22 April 2021, from https://www.who.int/emergencies/diseases/novel-coronavirus-2019/technical-guidance/naming-the-coronavirus-disease-(covid-2019)-and-the-virus-that-causes-it.

World Health Organisation. (2020). Coronavirus. Who.int. Retrieved 22 April 2021, from https://www.who.int/health-topics/coronavirus#tab=tab_1.

Garcia, K., & Berton, L. (2021). Topic detection and sentiment analysis in Twitter content related to COVID-19 from Brazil and the USA. Applied Soft Computing, 101(March), 1-15. https://doi.org/10.1016/j.asoc.2020.107057

Haupt, M., Jinich-Diamant, A., Li, J., Nali, M., & Mackey, T. (2021). Characterizing twitter user topics and communication network dynamics of the "Liberate" movement during COVID-19 using unsupervised machine learning and social network analysis. Online Social Networks And Media, 21(January), 1-14. https://doi.org/10.1016/j.osnem.2020.100114

Preda, G. (2021). All COVID-19 Vaccines Tweets. Kaggle.com. Retrieved 6 April 2021, from https://www.kaggle.com/gpreda/all-covid19-vaccines-tweets.

Stecanella, B. (2019). What is TF-IDF?. Monkeylearn.com. Retrieved 27 April 2021, from https://monkeylearn.com/blog/what-is-tf-idf/.

Goodman, T. (2014). Latent Dirichlet Allocation vs Hierarchical Dirichlet Process. Data Science Stack Exchange. Retrieved 27 April 2021, from https://datascience.stackexchange.com/questions/128/latent-dirichlet-allocation-vs-hierarchical-dirichlet-process).

Kumar, K. (2018). Evaluation of Topic Modeling: Topic Coherence. DataScience+. Retrieved 27 April 2021, from https://datascienceplus.com/evaluation-of-topic-modeling-topic-coherence/.

Sara. (2019). What is the meaning of Coherence score 0.4? Is it good or bad?. Stack Overflow. Retrieved 27 April 2021, from https://stackoverflow.com/questions/54762690/what-is-the-meaning-of-coherence-score-0-4-is-it-good-or-bad.

Pascual, F. (2019). Introduction To Topic Modeling. Monkeylearn.com. Retrieved 27 April 2021, from https://monkeylearn.com/blog/introduction-to-topic-modeling/.

MonkeyLearn. Monkeylearn.com. Retrieved 27 April 2021, from https://monkeylearn.com/sentiment-analysis/.

Li, S. (2018). Topic Modeling and Latent Dirichlet Allocation (LDA) in Python. Medium. Retrieved 27 April 2021, from https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24.

Liu, S. (2019). Dirichlet distribution. Medium. Retrieved 27 April 2021, from https://towardsdatascience.com/dirichlet-distribution-a82ab942a879.

Xu, J. (2018). Topic Modeling with LSA, PSLA, LDA & lda2Vec. Medium. Retrieved 27 April 2021, from https://medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05.

Sroka, E. C. (2020, May 27). Don't be Afraid of Nonparametric Topic Models (Part 2: Python). Medium. https://towardsdatascience.com/dont-be-afraid-of-nonparametric-topic-models-part-2-python-e5666db347a

Revert, F. (2019, September). An Overview of Topics Extraction in Python with Latent Dirichlet Allocation. KDnuggets. https://www.kdnuggets.com/2019/09/overview-topics-extraction-python-latent-dirichlet-allocation.html

Arvindpdmn, P. (2020, January 14). Topic Modelling. Devopedia. https://devopedia.org/topic-modelling

Managing a Commercial Greenhouse: Fuzzy Logic Controller

Georgios Chantziplakis
Faculty of Engineering, Environment, and Computing
MSc. Data Science and Computational Intelligence
Coventry University, Coventry, United Kingdom
Chantziplg@uni.coventry.ac.uk

Abstract - The aim of this paper is to demonstrate how to use fuzzy logic to control and regulate the environment, climate, and irrigation of commercial greenhouse. The input variables for our greenhouse are built on both internal and external factors like temperature, lighting, PH, humidity, soil moisture content, and cloudiness, further, based on the inputs our controllers react to keep the optimal environment for our smart greenhouse. The fuzzy logic controller simulation was created in MATLAB using the fuzzy logic toolbox, which is an inbuilt component. In the second part of this paper, we explained how the Genetic Algorithm works to obtain the most optimal set of rules for our controller and the differences between Mamdani and Sugeno. Finally, we compared two different optimization techniques, Genetic Algorithm and Particle Swarm Optimization using two functions from the CEC'2005 suite of benchmark functions.

Keywords – Fuzzy Logic Controller, Genetic Algorithm, Smart Greenhouse, Optimization

Introduction

Smart greenhouses are a revolution in agriculture using with sensors and actuators to monitor and control systems that optimise growth conditions and automate the growing process to create a self-regulating, micro-climate suitable for plant growth.
Between 2017 and 2022, the global smart greenhouse market is projected to expand at a CAGR of about 14.12 percent, from USD 680.3 million in 2016 to USD 1.31 billion in 2022. (Smart Greenhouse, 2020)
Smart greenhouses, which are equipped with advanced sensor and communications technologies, automatically collect, and distribute information on the environment and crop 24 hours a day, seven days a week. Data is collected and fed into an IoT platform, where analytical algorithms transform it into actionable information to identify bottlenecks and anomalies. The advantages are that helps growers to reduce labour costs, increase resource and chemical products, and increase yield rates by unlocking massive crop insights. (4 Benefits of Smart Greenhouses and How to Get Started | BehrTech Blog, 2020)

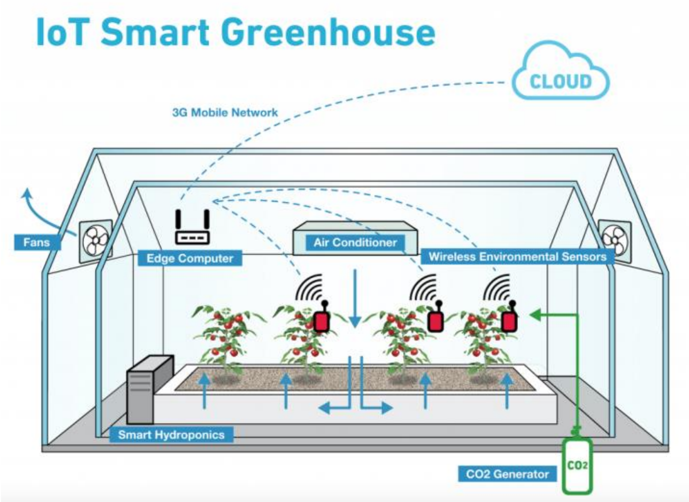


Figure 14 – Smart Greenhouse Example (Vieru, 2018)

Fuzzy Logic is a technique of reasoning that imitates human reasoning. The approach of Fuzzy Logic resembles the concept of decision making in humans, that considers all potential possibilities between digital values Yes or No, represented as 1.0 and 0.0, respectively. (Artificial Intelligence - Fuzzy Logic Systems - Tutorialspoint, n.d.)

We are tasked with designing, optimizing, and maintaining a Fuzzy Logic Controller (FLC) that manages a commercial smart greenhouse to control the climate and irrigation. There will be a variety of parameters used that will influence and assist the greenhouse. A demonstration of this would be temperature and fans to support the growth of the hot climate.

Literature Review

In the paper by Azaza et al. (2016), they use outputs as temperature, humidity, CO2, and illuminance to control the climate of the greenhouse dependable on the rules. A rule would be dependable on ventilation rate or heating rate, therefore depending on the scale of these ratings would activate the output of temperature more than illuminance as an example. The outputs and inputs are measured from very low, low, medium, high, and very high in exception of CO2 and shading which is measured from negative, zero, positive (CO2) and opened, half-opened and closed (Shading).

The consideration of outside temperature is taking into account as this can affect the inside temperature, and this would be something to consider when implementing the fuzzy logic controller of our own. Other rules this paper could have taken into consideration are the actual contents in the greenhouse for example soil content. We understand that the surroundings and inside of the greenhouse would need to be flawlessly developed however the contents inside are arguably a bigger priority ensuring the soil content is correctly monitored and sustained.

While this paper considers the majority of the important topics regarding greenhouse controlling, we believe a vital element of the contents inside the greenhouse is not taken into account as much as it should have been. (Azaza et al., 2016)

The paper published by Oubehar et al. (2020), closely studies and implements the use of Fuzzy Logic controller on a proposed real-life model of a climate controlling greenhouse. The authors wanted to understand the evolution of the greenhouse. The sensors used as inputs to detect the anomalies are temperature, humidity, and CO2 content. Any sensor that passes the threshold of anomaly would turn the heater fan (output) to return the greenhouse to its normal state.

With only 3 inputs and 1 output to assist the greenhouse with any issues, this number is too low and should take into considerations over parameters such as lighting as this is an important feature in a greenhouse for an input. In addition, a cooling fan or sprinkler to counter any abnormally hot conditions.

Part 1 - Designing and Implementing – FLC

Design

The design of the greenhouse is planned from research to implement the necessary inputs and outputs to allow a sustainable greenhouse. We intend on using 6 input sensors and 4 outputs controllers which include:

- Temperature (Input)
- Lighting (Input)
- Humidity (Input)
- Soil Moisture Content (Input)
- CO2 (Input)
- PH (Input)
- Heating (Output)
- Lamps (Output)
- Sprinklers (Output)
- Cooling Fans (Output)

The fuzzy logic system will be designed and implemented using MatLab Fuzzy Logic Toolbox. The below screenshot showcases the inputs and outputs.

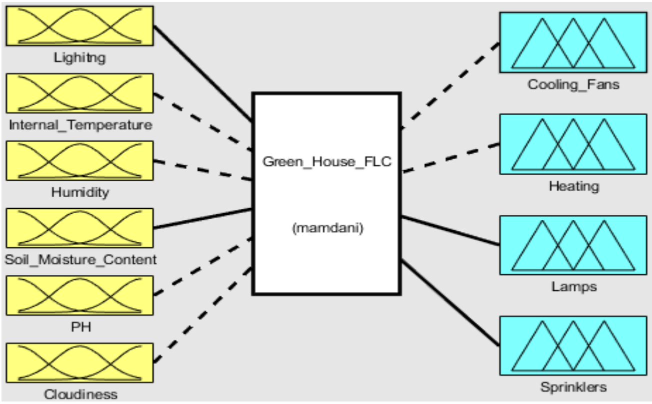


Figure 15 – Fuzzy Logic Controller (Inputs and Outputs)

The Fuzzy Inference System we plan on using is Mamdani as opposed to the Sugeno approach. We chose the Mamdani method because there were in our opinion far more advantages and ease of use. For example, the output membership function is present in contrast to the Sugeno method where this function is not present. The flexibility of Mamdani seems to be more lenient in terms of rules and inference process.

FLC Sensor Inputs and Outputs

There will be a total of 6 input sensors to manage the fuzzy logic controller. Each sensor has a specific output that if detected and dependable on the measurement, will activate the output variable to a specific velocity. The range of each output is detailed below:

- Temperature	-10 to 60	(Degrees Celsius)
- Lighting	0 to 2000	(Lux – 2% of 100000)
- Humidity	0 to 100	(Percentage)
- Soil Moisture Content	0 to 100	(Percentage)
- Cloudiness	0 to 100	(Percentage)
- PH	0 to 14	(PH scale Acid/Neutral/Alkaline)

For temperature, we decided on the range to be from -10 and 60 degrees Celsius due to the fact in the winter it can be below 0 and on the other hand in the summer it most likely will not go to 60, however allowing for this type of range would counter the worst-case scenario. Next, for Lighting, we chose to measure this on Lux. This measurement is the range of brightness, therefore 0 being dull and dark, and 1000 being the brightest and most light, 2000 being night and dark. We defined the range to be up to 2000 which is a small sample of the maximum lux range of ~100,000 which is based on a bright summers' day, in comparison to our example which 1000 is based on a bright summers day. (How to Measure Light Intensity, 2020). For Humidity, Soil moisture content, and Cloudiness we chose to measure the output based on percentage. For example, the sensor detecting 35% cloudiness would be classed as low. Lastly, for PH we measure this using the PH scale. More specifically, 0 to 6 is classified as an acid, 7 is defined as neutral, and 8 to 14 is classed as an alkaline. Ensuring that this figure is kept to 7 is important.

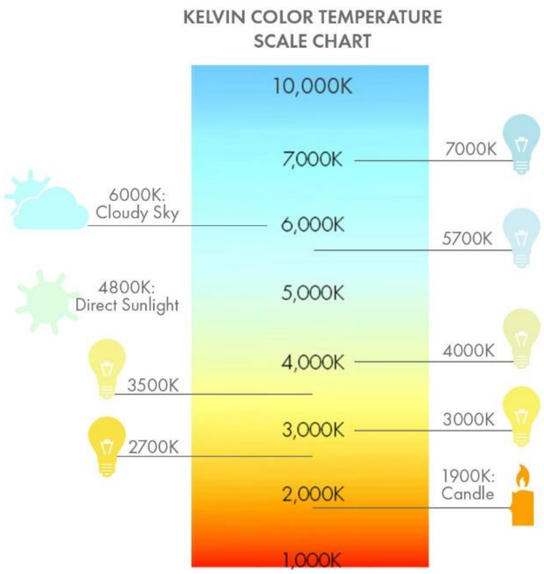


Figure 16 – Lumens Brightness Chart (Y, 2020)

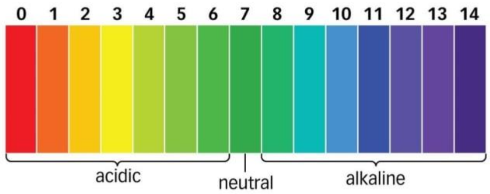


Figure 17 – PH Scale (PH Scale Defined - What Is PH?, n.d.)

Moving onto the outputs that can control the variables dependable on the measurement given, a range of each input is detailed below:

- Lamp 0 to 100 (Percentage)
- Heater 0 to 100 (Percentage)
- Sprinkler 0 to 100 (Percentage)
- Cooling fan 0 to 100 (Percentage)

The outputs are all measured from 0 – 100, using this scale allows us to easily capture the lower and higher measurements to categorise at what point the controllers are activated. The use of each output is to reduce the input returning the parameter to its original and optimal state. Below explains each input followed by the output it works with.

Lighting	→	Lamps
Heating	→	Heating & Cooling Fan
Humidity	→	Sprinkler & Cooling Fan
Soil Moisture Content	→	Sprinkler
PH	→	Heating & Cooling Fan
Cloudiness	→	Lamps

All the outputs work together where if multiple inputs are experiencing a low or high measurement, then the required outputs will be activated, and this can be evidenced from scenario analysis.

Fuzzy Logic Inference

Fuzzy inference is a technique that interprets the values from the input vector, built on a variety set of rules. Most commonly used fuzzy inference method is Mamdani, due to its simple and easy to use method. It began as a method to produce a control system by synthesizing a set of linguistic control rules, gathered by experienced human operators. (Mamdani and Sugeno Fuzzy Inference Systems - MATLAB & Simulink - MathWorks United Kingdom, n.d.)
In order to implement the fuzzy inference Mamdani method, first, the operator should create a set of fuzzy rules that apply to each input then the fuzzification process begins, where the input values are fuzzified using the input membership functions. Further, the inputs that have been fuzzified will be

combined according to the fuzzy rules made. Then dependable on the determined result will get a selection from the combination of outputs that have been specified from the fuzzy rules according to the membership function.

Fuzzy Input Combination

After we have defined the membership functions for each input sensor, the fuzzy rules are then created and defined for each situation. Ensuring that each rule is completed for combination uses using the and/or statements. The fuzzy rules will follow format: *IF [Inputs], and/or, THEN [Outputs]*. You can use multiple inputs/outputs for these rules to cover all basis. Over 200 rules were produced and defined to ensure the rules were fit for each output in different environmental conditions the greenhouse can encounter. The example below showcases the use of multiple inputs and outputs being used in the if statement to accommodate for that scenario. The fis file with all the rules included can be found in Appendix.

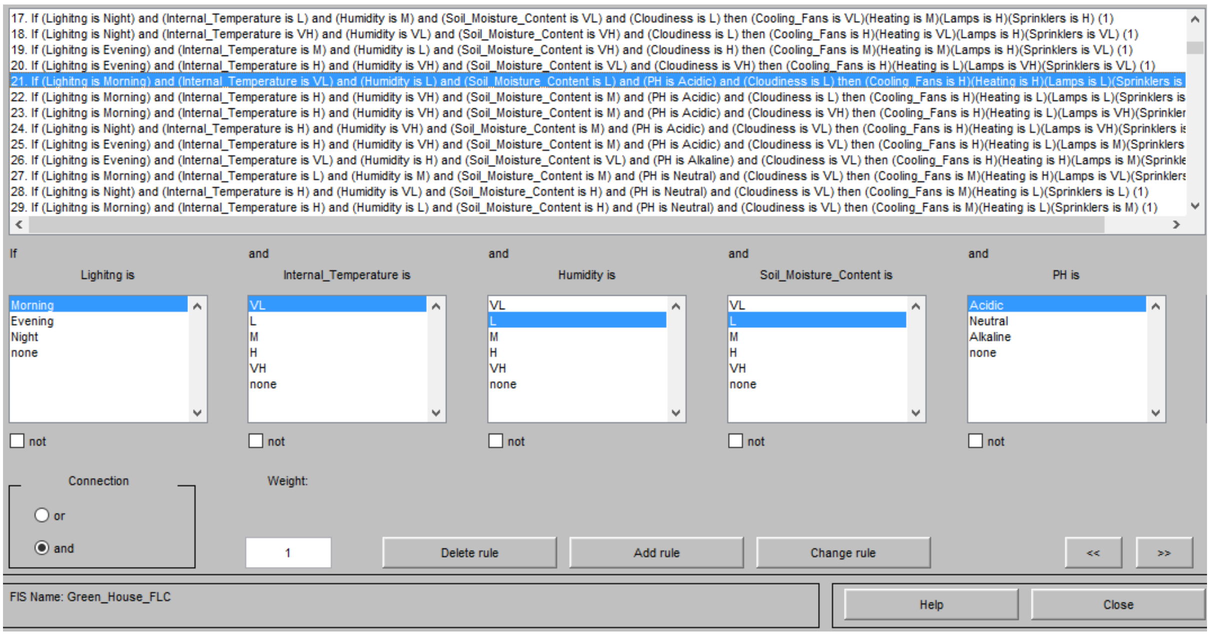


Figure 18 – Implemented Rules in MATLAB

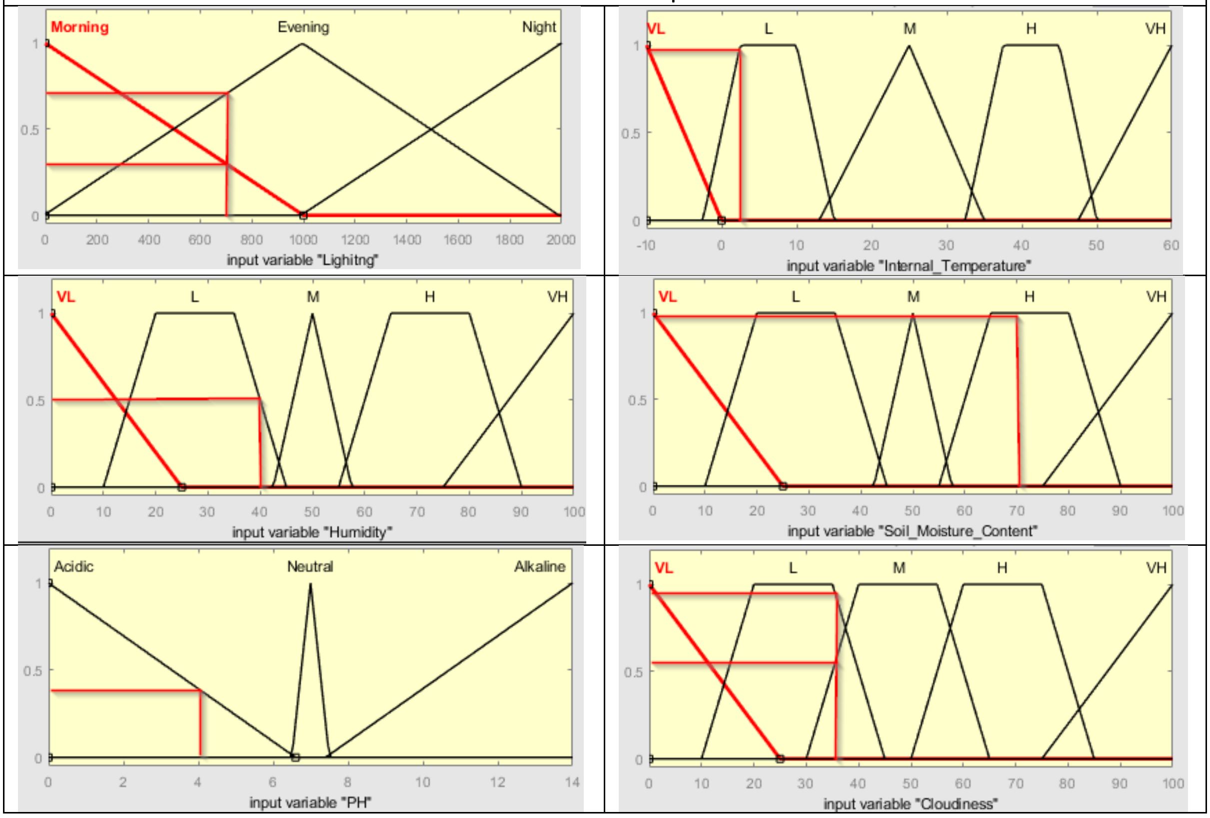
Table 2 – Example of Rules

Output/Input	Rule 1	Rule 2	Rule 3
Lighting	Evening	Evening	Morning
Temperature	L	N/A	VL
Humidity	L	L	VL
Soil Moisture Content	H	M	N/A
PH	Acidic	Alkaline	Acidic
Cloudiness	L	N/A	N/A
Lamps	L	M	VL
Heating	M	H	H
Sprinkler	L	M	VL
Cooling Fan	L	H	L

Key: VL = Very low, L = Low, M = Medium, H = High, VH = Very High, N/A = Not Applicable

Table 3 – Membership Function of Each Input

Rule 1 – Example



From the membership functions we evaluate the results using table 4, this demonstrates that the minimum value input variable (highlighted in blue) is PH of 0.4 which reflects the outputs (highlighted in green).

Table 4 – Membership Function Evaluation Results

Rule 1 Membership Function Evaluation - Example		
Input/Output	Variable Value	Evaluation Result
Lighting	700	0.75
Temperature	1	0.95
Humidity	40	0.5
Soil Moisture Content	70	1
PH	4	0.4
Cloudiness	35	0.95
Lamps		0.4
Heaters		0.4
Sprinklers		0.4
Cooling Fans		0.4

Defuzzification

The method of obtaining a single number from the output of an aggregated fuzzy set is known as defuzzification. It is used to convert the effects of fuzzy inference into a crisp output. In simple words, defuzzification is accomplished through a decision-making algorithm that selects the best crisp value from a fuzzy set. The center of gravity (COG), mean of maximum (MOM), and center average methods are all examples of defuzzification. The COG method returns the value of the curve's center of area under the curve, while the MOM method can be thought of as the point where a curve's equilibrium is achieved. (Masoum & Fuchs, 2015)

Relating this to our fuzzy logic controlling model, the COG is derived by calculating the area under the curves of the aggregate outputs. Furthermore, simplifying the computation, the area will be calculated by presuming miniscule rectangles are filling the space. In addition, the area would be the width of the x axis multiplied by the length of the rectangle for each respective rectangle. In the final step, it gathers all the output rules, and these are aggregated and defuzzification is applied. It transforms the fuzzy rule outputs into output values that will be received by the actuators.

The formula for the COG method can be seen below:

$$COG[C(w)] = \frac{\sum_{i=1}^{q} w_i C(w_i)}{\sum_{i=1}^{q} C(w_i)}$$

Figure 19 - Centre of Gravity Defuzzification Formula (Negnevitsky, 2011)

The above formula's performance will be sent to the sensors, to make all the necessary changes/actions according to our rules. The MATLAB toolbox graphical representation of the aggregated output on a couple of rules for some inputs is shown in the figure below.

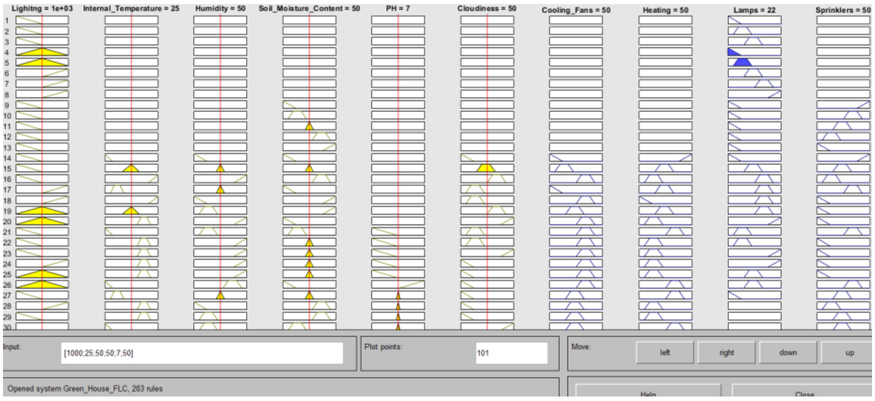


Figure 20 – Aggregated Outputs of Rules

Finally, using MATLAB, surface plots can determine the relationships between two inputs and an output. The plot below shows the relationship between ‘lighting’ and ‘cloudiness’ on ‘lamps.’

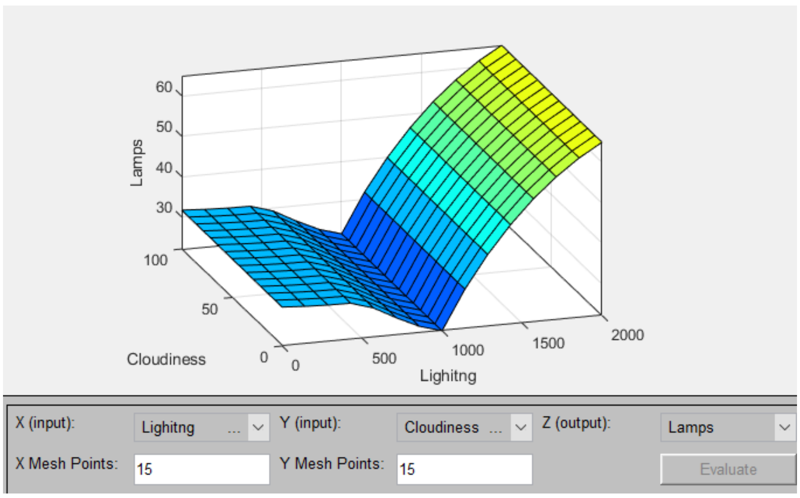


Figure 21 – Surface Plot Relationship Between Inputs Lighting and Cloudiness on Lamps

Scenario Analysis

The following scenarios cover the main season of winter and summer periods which also cover the 3 lighting periods of morning, evening, and night. Therefore 6 main scenarios are being analysed in this section, where we take into consideration the time of each season along with the sunlight.

Winter Season Scenario

Table 5 – Winter Season Scenarios

Output/Input	**Scenario 1**	**Scenario 2**	**Scenario 3**
Lighting	Morning	Evening	Night
Temperature	L	L	VL
Humidity	M	H	VL
Soil Moisture Content	L	M	H
PH	Acidic	Neutral	Acidic
Cloudiness	H	VL	L
Lamps	M	VL	H
Heating	L	VL	M
Sprinkler	M	M	VL
Cooling Fan	L	VL	L

Key: VL = Very low, L = Low, M = Medium, H = High, VH = Very High

In table 4, different scenarios are illustrated that maybe encountered during a winter season, these scenarios attempt to tackle and investigate if the weather is rainy, windy or sunny. During the winter season the temperature is predominately low and cold hence the scale of temperature being between VL to L which covers the area of -10 degrees to 15 degrees. Moving onto humidity and dependable on if it rains humidity is mostly high which is the case for scenario 2. Different soil moisture contents are investigated to understand the if the soil is simply dry (VL/L) or wet (H/VH). For the winter season, the likelihood that the soil moisture will be very dry is low due to the bad weather conditions. Next is the PH and the optimal measurement is 7 (neutral). This can affect the growth of the contents in the greenhouse, therefore if this is acidic or alkaline then the outputs are required to assistance bring the scale back to its ideal measurement. Lastly, cloudiness on a rainy day or the morning due to fog will be high however throughout the day this can clear itself but can help with the use of lamps to increase visibility.

Summer Season Scenario

Table 6 – Summer Season Scenarios

Output/Input	**Scenario 1**	**Scenario 2**	**Scenario 3**
Lighting	Morning	Evening	Night
Temperature	L	M	L
Humidity	VL	H	L
Soil Moisture Content	L	VL	L
PH	Acid	Alkaline	Neutral
Cloudiness	L	VL	VL
Lamps	L	M	H
Heating	M	H	L
Sprinkler	H	H	M
Cooling Fan	M	VH	L

In table 5, examines the summer scenarios which covers morning evening and night, which also considers different weather types for each scenario. Firstly, for temperature the use of M and L are

used however the use of H could be implemented in the summer given the range of temperature from -2.5 to 50 degree which is a very large area to cover however the likelihood of the greenhouse reaching this degree is low but cannot be disregarded. In addition, as a greenhouse is an enclosed area the temperature can increase rapidly and reaching over 30 degrees on a hot day, however this will most likely only happen in the evening on a summer's day. Next, we look at humidity and this can be high when the sun is out however can be regarded as low during the morning and night. Due to the higher temperatures in the summer, mostly the soil moisture will be dry (VL/L) and will need the output sprinklers to assist. Onto PH, this can vary and does not rely on any season/time of year to be predominately a certain kind of PH. However, the optimal and best PH is neutral therefore always aiming to keep this as 7 (neutral). Lastly, we look at cloudiness and this will be low due to the less likelihood of rain and fog during this time of year.
Scenario 1 from the summer season, is used as an example in MATLAB as evidence of the system working. The whole MATLAB can be found in the Appendix.

```
>> FLC_Evaluation
What is the lighting from 0 to 2000: 100
What is the internal temperature from -10 to 60: 1
What is the humidity from 0 to 100: 10
What is the soil moisture from 0 to 100: 20
What is the ph from 0 to 14: 4
What is the cloudiness from 0 to 100: 11
Turn fans on Medium
Turn heater on Medium
Turn lamps on Low
Turn sprinklers on High
```

Figure 22 – FLC Working Scenario in MATLAB

Part 2 - Optimize the Developed FLC

After developing our fuzzy logic controller for greenhouse climate and irrigation control using rules, we will try to improve them and get the best outcome by using a Genetic Algorithm (GA). A genetic algorithm is a search heuristic based on Charles Darwin's natural selection theory. This algorithm mimics natural selection, in which the fittest individuals are chosen for reproduction to produce the next generation's offspring. The process of natural selection starts with the selection of the fittest individuals from a population. They produce offspring that inherit the parents' characteristics and are passed on to the next generation. If parents are physically fit, their children would be fitter than they are and have a greater chance of survival. This mechanism will continue to iterate until a generation of the fittest individuals is discovered.
The phase starts with a set of Individuals known as a Population. Each Individual is a potential solution to the problem we are trying to solve. Genes are a collection of parameters (variables) that define an Individual. A Chromosome is made up of a string of genes (solution). The set of genes of an Individual is expressed by a string in terms of an alphabet in a genetic algorithm. Binary values are commonly used (a string of 1s and 0s). (Mallawaarachchi, 2017)

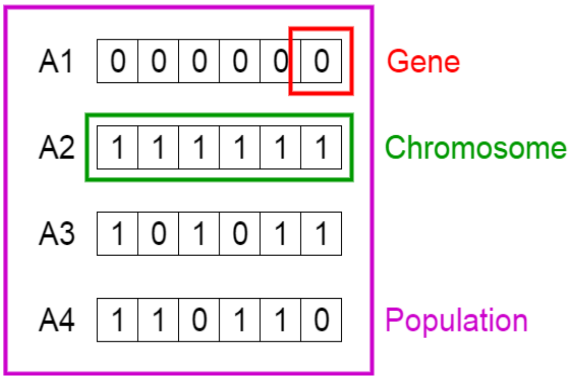


Figure 23 - GA Representation

Design

In this section, we will demonstrate the steps of optimizing our FLC from task 1 using GA. The controller's inputs and outputs should be converted to binary values, for the first step before the genetic algorithm commences which is transformed into a binary integer the length of 8.
Our FLC contains Trapezoid (Tp) and Triangular (Tg) Membership Functions. This means that for optimization, each trapezoid Membership Function has four parameters, and each triangular function has three. The total parameters to be optimized are 154 and given the number of parameters there are 1232 genes in total. More specifically for the parameters we have 86 (17 + 17 + 17 + 17 + 9 + 9) from our Inputs and 68 (17 + 17 + 17+ 17) from our Outputs.
The following tables indicate the number of genes calculated for each input and output, as well as the representation of each. Note that we can only represent the first two digits of the number in light binary representation since the light range is from 0 to 2,000 lux (Table 10).

Input Tables

Table 7 - Input Variable Temperature

Internal Temperature	Value	Binary (8-bit)
Very Low	-10	11110110
	-10	11110110
	0	00000000
Low	-2.5	11111110
	2.5	00000010
	10	00001010
	15	00001111
Medium	13	00001101
	25	00011001
	35	00100011
High	32.5	00100000
	37.5	00100101
	45	00101101
	50	00110010
Very High	47.5	00101111
	60	00111100
	60	00111100

Table 8 – Input Variable Humidity

Humidity	Value	Binary (8-bit)
Very Low	0	00000000
	0	00000000
	25	00011001
Low	10	00001010
	20	00010100
	35	00100011
	45	00101101
Medium	42.5	00101010
	50	00110010
	57.5	00111001
High	55	00110111
	65	01000001
	80	01010000
	90	01011010
Very High	75	01001011
	100	01100100
	100	01100100

Table 9 – Input Variable Soil Moisture Content

Soil Moisture Content	Value	Binary (8-bit)
Very Low	0	00000000
	0	00000000
	25	00011001
Low	10	00001010
	20	00010100
	35	00100011
	45	00101101
Medium	42.5	00101010
	50	00110010
	57.5	00111001
High	55	00110111
	65	01000001
	80	01010000
	90	01011010
Very High	75	01001011
	100	01100100
	100	01100100

Table 10 – Input Variable Cloudiness

Cloudiness	Value	Binary (8-bit)
Very Low	0	00000000
	0	00000000
	25	00011001
Low	10	00001010
	20	00010100

	35	00100011
	45	00101101
Medium	42.5	00101010
	50	00110010
	57.5	00111001
High	55	00110111
	65	01000001
	80	01010000
	90	01011010
Very High	75	01001011
	100	01100100
	100	01100100

Table 11 – Input Variable Lighting

Lighting	Value	Binary (8-bit)
Morning	0	00000000
	0	00000000
	1000	01100100
Evening	-5	11111011
	995	01100011
	1990	00010011
Night	990	01100011
	2000	00010100
	2000	00010100

Table 12 – Input Variable PH

PH	Value	Binary (8-bit)
Acidic	0	00000000
	0	00000000
	6.5	00000110
Neutral	6.5	00000110
	7	00000111
	7.5	00000111
Alkaline	7.5	00000111
	14	00001110
	14	00001110

Output Tables

Table 13 – Output Variable Cooling Fans

Cooling Fans	Value	Binary (8-bit)
Very Low	0	00000000
	0	00000000
	25	00011001
Low	10	00001010
	20	00010100

	35	00100011
	45	00101101
Medium	42.5	00101010
	50	00110010
	57.5	00111001
High	55	00110111
	65	01000001
	80	01010000
	90	01011010
Very High	75	01001011
	100	01100100
	100	01100100

Table 14 – Output Variable Heating

Heating	Value	Binary (8-bit)
Very Low	0	00000000
	0	00000000
	25	00011001
Low	10	00001010
	20	00010100
	35	00100011
	45	00101101
Medium	42.5	00101010
	50	00110010
	57.5	00111001
High	55	00110111
	65	01000001
	80	01010000
	90	01011010
Very High	75	01001011
	100	01100100
	100	01100100

Table 15 – Output Variable Lamps

Lamps	Value	Binary (8-bit)
Very Low	0	00000000
	0	00000000
	25	00011001
Low	10	00001010
	20	00010100
	35	00100011
	45	00101101
Medium	42.5	00101010
	50	00110010
	57.5	00111001
High	55	00110111
	65	01000001
	80	01010000

	90	01011010
Very High	75	01001011
	100	01100100
	100	01100100

Table 16 – Output Variable Sprinklers

Sprinklers	Value	Binary (8-bit)
Very Low	0	00000000
	0	00000000
	25	00011001
Low	10	00001010
	20	00010100
	35	00100011
	45	00101101
Medium	42.5	00101010
	50	00110010
	57.5	00111001
High	55	00110111
	65	01000001
	80	01010000
	90	01011010
Very High	75	01001011
	100	01100100
	100	01100100

The transformation to binary was complete with the assistance of Math is Fun website. (Binary/Decimal/Hexadecimal Converter, n.d.)

Genetic Algorithm Steps

The crossover probability and the mutation probability must be chosen after each membership function for inputs and outputs that have been represented using binary representation.
Crossover is a genetic operation in which two parent chromosomes are combined to create two new offspring chromosomes. The theory behind crossover is that if the new chromosome inherits the best traits from both parents and so it might be better than both. During evolution, crossover occurs according to a user-defined crossover likelihood. The crossover likelihood is usually between 0 and 1, with a value of 0.7 producing good results.
The mutation probability, on the other hand, is much smaller and is held between 0.001 and 0.01 to avoid being trapped at a local optimum. Only a few individuals are impacted by mutation. A random number between zero and one is produced for each gene in each chromosome to search for potential mutations. The gene value is modified if this number is less than or equal to the given mutation probability, i.e. 0.01. Mutations provide diversity, allowing researchers to search in domain regions that would otherwise be overlooked. (Pawar & Bichkar, 2015)

The optimization process will begin once the chromosome size, fitness function, termination requirements, crossover, and mutation parameters have all been determined.

Step 1:

A population with a randomly chosen number of chromosomes is initialised, with each gene in the chromosome being randomly assigned a value with no order. We will choose four chromosomes at random after determining the length of the chromosome just to demonstrate crossover and mutation. For demonstration purposes, we will choose our chromosome to be in size of 10.

0	1	2	3	4	5	6	7	8	9

-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

25	26	27	28	29	30	31	32	33	34

52	53	54	55	56	57	58	59	60	61

Step 2:

A data collection of values will be transferred into the chromosome. For each chromosome, the MSE (mean squared error) is determined by comparing it to a fitness function that is defined. It is worth noting that the chromosomes mentioned above should be decoded in binary form so that they can be evaluated.

Step 3:

We compare the MSE values to the stopping criteria. If the optimum value is not met, the process will proceed.

Step 4:

The chromosomes must be chosen for mating. To find chromosomes for mating, we use the Roulette Wheel method of Selection (RWS).

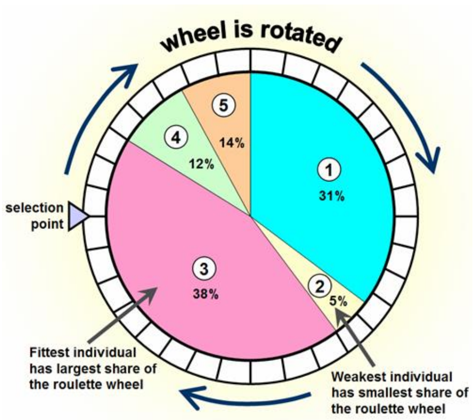


Figure 24 - Roulette Wheel Selection Representation

An example of how it works is that if we imagine that the MSE of the first chromosome was the lowest, followed by the second, third, and fourth. The wheel space ratios for each are determined, a demonstration of this can be evidenced in Figure 24. This means that the first chromosome has a much higher probability of being chosen for mating in comparison with the fourth chromosome.

The wheel would be rotated as many times as there are chromosomes (in this case 4 times). After spinning, the first and second will mate, and the third and fourth will mate. Second and third spins picked third and second respectively and finally the fourth spin selected the first. Only the best chromosomes survive, so the fourth chromosome that was not chosen would be deleted.

Step 5:

We now pick first and third chromosomes for mating based on a crossover probability of 0.7. The split point style matting is chosen at random. A demonstration can be seen below.

Before crossover:

0	1	2	3	4	5	6	7	8	9

25	26	27	28	29	30	31	32	33	34

After crossover:

25	26	27	28	29	5	6	7	8	9

0	1	2	3	4	30	31	32	33	34

The mated chromosome one with chromosome two is treated in the same way. So now following the same process we will have:

Before crossover:

25	26	27	28	29	5	6	7	8	9

-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

After crossover:

-10	-9	-8	-7	-6	5	6	7	8	9

25	26	27	28	29	-5	-4	-3	-2	-1

Step 6:

Following crossover, the mutation process occurs with an exceptionally low probability which picks a gene at random and mutates it. It is unlikely to occur often due to the low likelihood. So, we will have that:

Before Mutation:

-10	-9	-8	-7	-6	5	6	7	8	9

25	26	27	28	29	-5	-4	-3	-2	-1

After Mutation:

-10	-9	-8	-7	-6	23	6	7	8	9

25	26	27	-33	29	-5	-4	-3	-2	-1

After all these steps, a new population is created, and the newly formed chromosomes are shown below.

-10	-9	-8	-7	-6	23	6	7	8	9

25	26	27	-33	29	-5	-4	-3	-2	-1

25	26	27	28	29	5	6	7	8	9

0	1	2	3	4	30	31	32	33	34

Step 7:

We verify that the new population's chromosome length does not surpass the chromosome length. This procedure will be repeated until the best MF has been chosen.

FLC Rule Optimization

Fuzzy Logic is formed based on the input and output expectations and the membership functions are used to construct the rules. However, those rules do not ensure that our system will return the most optimal results. We use a genetic algorithm to solve this problem that assists us in determining the most optimal solution. The total number of possible rules must be determined before the fuzzy rule optimization process can begin. This is accomplished by multiplying the number of input and output membership functions.

Table 17 - Number of Membership Functions of each Input and Output

Lighting	3
Internal Temperature	5
Humidity	5
Soil Moisture Content	5
PH	3
Cloudiness	5
Cooling Fans	5
Heating	5
Lamps	5
Sprinklers	5

So, the total numbers of rule in our example are 3*5*5*5*3*5*5*5*5*5 = 3,515,625. Taking the product of the inputs alone will give us the specific rules and prevent overlap. As a result, the total number of rules is: 3*5*5*5*3*5 = 5625. Individual outcomes must be calculated based on the outputs, which in that case we have 5*5*5*5*5 = 3125.

The rules are written in a fuzzy grid of statements in the form of ‘IF-THEN’. The following is a list of fuzzy rules that correspond to each other:

$$R_{(j_1,j_2,\ldots,j_n)} : \text{IF } x_1 \text{ is } A_{(1,j_1)} \text{ and } x_2 \text{ is } A_{(2,j_2)} \text{ and}, \ldots, x_n \text{ is } A_{(n,j_n)},$$
$$\text{THEN } \overline{x} \text{ belong to Class } y_{(j_1,j_2,\ldots,j_n)} \text{ with } CF = CF_{(j_1,j_2,j_n)},$$

Figure 25 - Fuzzy Rules Representation (Mao et al., 2020)

Then the outputs are mapped to all the established fuzzy rules and the weights of each output determine the strength of each class. After finding the strength of all the classes we keep the maximum one and the GA is ready to be implemented.

Optimal number of rules

The final model with the best rules will have a high accuracy but also a high computational cost. Therefore, less rules should be used, and a genetic algorithm would be used to accomplish this. This algorithm will follow the same steps as the previous one, but with a different fitness function. So, to increase the accuracy and reduce the number of rules weights will be added to the fitness feature. As a result, the fitness function will be determined following the formula below.

$$f(S) = w_P \frac{P_s}{P_{ALL}} - w_N \frac{N_S}{N_{ALL}}$$

Figure 26 – Fitness Function Formula (Negnevitsky, 2011)

where Ps is the number of patterns classified successfully, P_{ALL} is the total number of patterns presented to the classification system, N_S and N_{ALL} are the numbers of fuzzy IF-THEN rules in set S and set S_{ALL}, respectively.

Mamdani vs. Sugeno

The model that we created was a Mamdani model. The overall length of the chromosome would be different if this was a Sugeno model, and it would be much less. In Mamdani, we used Membership Functions shapes like Trapezoid and Triangle, but we use zmf (Z-shaped membership function) for Sugeno. The shapeshift will be applied to both the inputs and outputs. Sugeno's outputs are interpreted as constants or linear functions of the input values since it uses a singleton output membership function. In general, the biggest difference would be that the total number of genes in the chromosome will decrease. (Mamdani and Sugeno Fuzzy Inference Systems - MATLAB & Simulink - MathWorks United Kingdom, n.d.)

Part 3 - Comparing Different Optimization Techniques on CEC'2005 Functions

In this part, we will compare the output of two different optimization techniques from two functions chosen from the CEC'2005 functions. There are 25 benchmark functions that come in a variety of forms, including unimodal, multimodal, and hybrid. (P. N. Suganthan et al., 2005). The precision of each optimization technique in getting to the global optimum or near to it will be the primary criterion for judging results.

Opimization Techniques

In terms of optimization techniques, we selected:

- ***Genetic Algorithm***

To recap, Genetic Algorithm is a technique for resolving optimization issues where the fittest individuals are selected for reproduction to produce the next generation's offspring, as in natural selection. The phase starts with a Population, which is a group of individuals. Each individual represents a potential solution to the problem we're trying to solve. Genes are a collection of parameters (variables) that characterise an individual. A collection of genes is referred to as a chromosome (solution). GA was explained analytically in the previous section.

- ***Particle Swarm Optimization***

Particle swarm optimization (PSO) is a population-based optimization method inspired by the behaviour of flocks of birds and schooling fish. PSO and evolutionary computing methods have a lot in common. The method starts with a population of random solutions, and the quest for the best one is done by updating generations. (Yang, 2020)

CEC'2005 Functions

The two functions chosen from the CEC'2005 are:

- ***Shifted Rotated Weierstrass Function (Function 11)***

$$F_{11}(\mathbf{x}) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k\max}[a^k \cos(2\pi b^k (z_i + 0.5))]\right) - D\sum_{k=0}^{k\max}[a^k \cos(2\pi b^k \cdot 0.5)] + f\_bias_{11},$$

a=0.5, b=3, k_{max}=20, $\mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M}$, $\mathbf{x} = [x_1, x_2, \ldots, x_D]$

D: dimensions

$\mathbf{o} = [o_1, o_2, \ldots, o_D]$: the shifted global optimum

M: linear transformation matrix, condition number=5

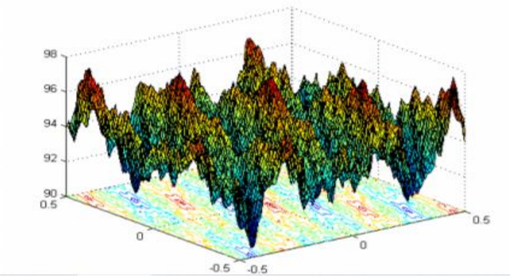


Figure 27 - Shifted Rotated Weierstrass

Properties:

- ❖ Multi-modal
- ❖ Shifted
- ❖ Rotated
- ❖ Non-separable
- ❖ Scalable
- ❖ Continuous but differentiable only on a set of points
- ❖ x ∈ [-0.5,0.5]D, Global optimum x* = 0, F11(x*) = f_bias11 = 90

- ***Shifted Rotated Expanded Scaffer's F6 Function (Function 14)***

$$F(x, y) = 0.5 + \frac{(\sin^2(\sqrt{x^2 + y^2}) - 0.5)}{(1 + 0.001(x^2 + y^2))^2}$$

Expanded to

$$F_{14}(\mathbf{x}) = EF(z_1, z_2, \ldots, z_D) = F(z_1, z_2) + F(z_2, z_3) + \ldots + F(z_{D-1}, z_D) + F(z_D, z_1) + f\_bias_{14},$$

$\mathbf{z} = (\mathbf{x} - \mathbf{o}) * \mathbf{M}$, $\mathbf{x} = [x_1, x_2, \ldots, x_D]$

D: dimensions

$\mathbf{o} = [o_1, o_2, \ldots, o_D]$: the shifted global optimum

M: linear transformation matrix, condition number=3

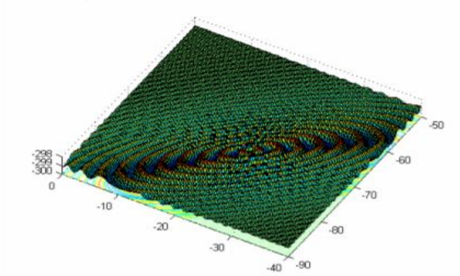


Figure 28 - Shifted Rotated Expanded Scaffer

Properties:

- ❖ Multi-modal
- ❖ Shifted
- ❖ Non-separable
- ❖ Scalable
- ❖ x ∈ [-100, 100] D, Global optimum x* = 0, F14(x*) = f_bias14 = -300

Each algorithm was run 15 times for more optimal performance, with the results of each run being registered. 2 Dimensions and 10 Dimensions are used to determine algorithms. In the tables below the best performance (lowest minima), the worst performance (highest minima), median value, standard deviation, and the number of generations/iterations are illustrated.

Function Results

To ensure rigorous testing, we run each algorithm 15 times and record the outputs from other functions and dimensions. The outputs records are Best Performance, Worst performance, Mean average, Standard deviation, and Best number of iterations. The best performance is calculated by identifying the lowest minima and for the worst performance, this is calculated by locating the highest minima.

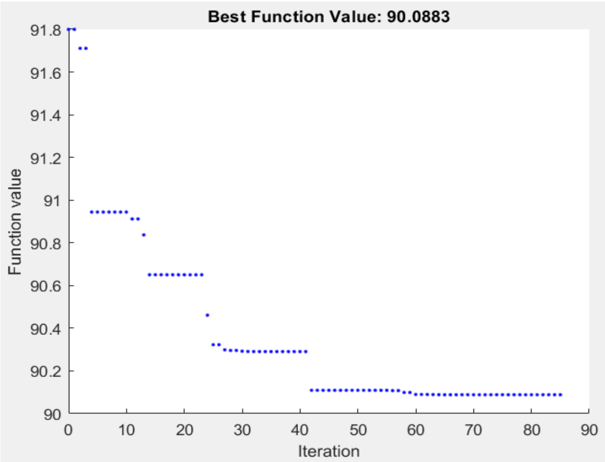


Figure 29 - Particle Swarm Optimization Function 11 (2-D)

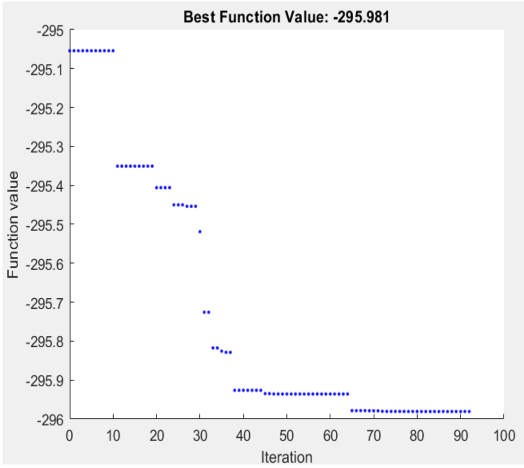


Figure 30 - Particle Swarm Optimization Function 14 (10-D)

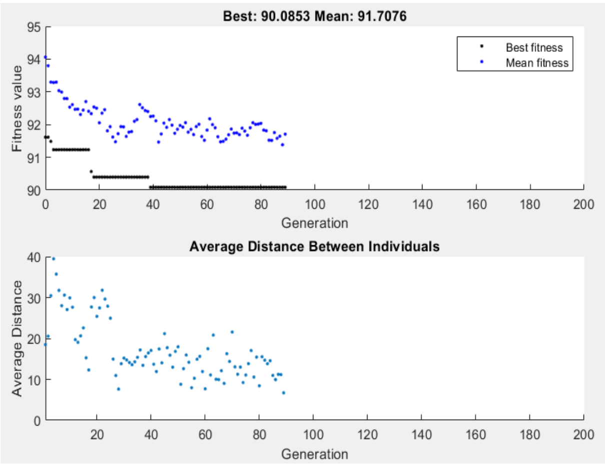


Figure 31 - Genetic Algorithm Function 11 (2-D)

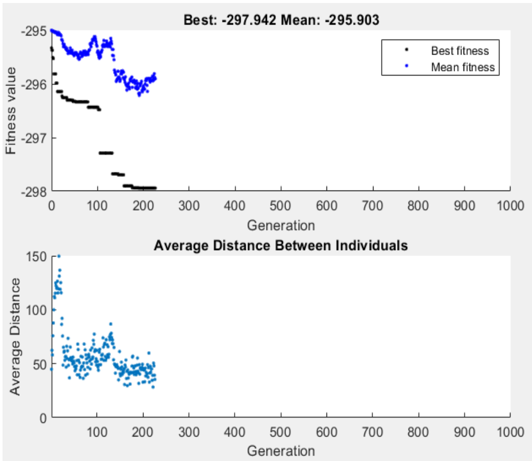


Figure 32 - Genetic Algorithm Function 14 (10 - D)

In the figures above, some iterations of the best functions are presented as an example. All the figures are saved and can be viewed in the Appendix.

The below tables are a summary of each algorithm's respective functions. Further, the full code and results and be evidenced in the appendix.

Table 18 - Summary of function 11 (2-D)

Shifted Rotated Weierstrass Function – Function 11 – 2D					
Algorithm	Best Performance	Worst Performance	Mean	Standard Deviation	Best number of Iterations
Genetic Algorithm	90.08	91.03	90.56	0.26	89
Particle Swarm	90.08	90.59	90.35	0.16	85

Examining the results between both algorithms for function 11, we find that both algorithms produce the same best result of 90.08 however Particle Swarm Optimization (PSO) does so in less generations. In addition, Genetic Algorithm (GA) gives ever so slightly worse results.

Table 19 - Summary of function 11 (10-D)

Shifted Rotated Weierstrass Function – Function 11 – 10D					
Algorithm	Best Performance	Worst Performance	Mean	Standard Deviation	Best number of Iterations
Genetic Algorithm	98.72	101.07	100.05	0.65	103
Particle Swarm	98.11	101.90	100.70	1.08	87

From the table it is clear that PSO was the more superior algorithm with its best performance bring 98.11 with only 87 iterations in comparison to the GA showing at 98.72 with 103 generations. It can be argued that GA provides more consistent results with the mean of the minima and standard deviation being lower.

Table 20 - Summary of function 14 (2-D)

Shifted Rotated Expanded Scaffer's F6 Function – Function 14 – 2D					
Algorithm	Best Performance	Worst Performance	Mean	Standard Deviation	Best number of Iterations
Genetic Algorithm	-299.98	-299.92	-299.96	0.01	91
Particle Swarm	-300.00	-299.92	-299.97	0.02	81

This function for PSO hit the targeted score of -300 with 81 iterations however GA was only 0.02 away which is with respect a good score considering but achieved this in more iterations. Therefore, PSO being the better algorithm in this situation contrasting between the two.

Table 21 - Summary of function 14 (10-D)

Shifted Rotated Expanded Scaffer's F6 Function – Function 14 – 10D					
Algorithm	Best Performance	Worst Performance	Mean	Standard Deviation	Best number of Iterations
Genetic Algorithm	-297.94	-296.29	-296.72	0.44	226
Particle Swarm	-295.98	-295.37	-295.51	0.14	92

This last experiment for function 14 for 10 dimensions showcased that GA performed significantly better than PSO achieving almost 2.0 in minima.

Ethical, Legal, Social Issues

Regarding ethical concerns, there are no major nor minor issues that require to be investigated for a smart automated greenhouse. A problem that does need to be considered is the safety of the equipment being used to ensure that the systems are providing the correcting output or not overheating. To counter this, yearly quality checks on the machinery used would be required and a necessity to be able ensure a successful smart greenhouse.

Discussion and Conclusion

To conclude, a full understanding of fuzzy logic is required to calculate the optimal rules and parameters. This paper serves as an extensive research into fuzzy logic focusing on smart greenhouse clime control. Mamdani Fuzzy logic inference was used, along with 6 inputs and 4 output variables. The inputs accommodate for the temperature, humidity, lighting, Soil moisture content, PH, and cloudiness and the outputs that support them are heaters, colling fans, sprinklers, and lamps. We evaluated the performance of the controller system. In addition, we adjusted the membership functions of each input and output variable by assigning them Trapezoid or Triangular types, defining the range and parameters of each variable.

In the second part of the work, we concentrated on investigating the use of Genetic Algorithm to find the optimal solution for our controller. The components implemented were fitness function, problem encoding and genetic operators. Furthermore, the Mamdani model was used over the Sugeno because of its advantages and capabilities it has to offer.

For part 3, we investigate the genetic algorithm and particle swarm optimization techniques on 2 functions from CEC'2005 functions. Experimentations were implemented on functions 11 and 14, known as Shifted Rotated Weierstrass Function and Shifted Rotated Expanded Scaffer's F6 Function, respectively. After investigation, it was found that Particle Swarm Optimization was well-suited for the 2 dimensions however it is difficult to differentiate between the 2 functions for 10 dimensions, therefore in this situation we regard both functions to be unsuitable for 10-dimension experiments.

Lastly, fuzzy logic can be applied to a variety of applications, the example of a smart commercial greenhouse is an excellent demonstration on how to do so.

Appendix

Link to code, figures and data:
https://1drv.ms/u/s!Ajl5nOLsDGcBgopKCtIUIO4ZLrRicQ?e=kQUdyY

References

Oubehar, H., Selmani, A., Ed-dahhak, A., Archidi, M., Lachhab, A., & Bouchikhi, B. (2020). Fuzzy-Super Twisting Sliding Mode for Greenhouse Climate Control. 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), 1–4. https://doi.org/10.1109/iraset48871.2020.9092127

Azaza, M., Tanougast, C., Fabrizio, E., & Mami, A. (2016). Smart greenhouse fuzzy logic based control system enhanced with wireless data monitoring. ISA Transactions, 61, 297–307. https://doi.org/10.1016/j.isatra.2015.12.006

4 Benefits of Smart Greenhouses and How to Get Started | BehrTech Blog. (2020, October 28). BehrTech. https://behrtech.com/blog/4-benefits-of-smart-greenhouses-and-how-to-get-started/

Smart greenhouse. (2020, November 4). Designing Buildings Wiki. https://www.designingbuildings.co.uk/wiki/Smart_greenhouse#:%7E:text=The%20smart%20greenhouse%20is%20a,and%20automate%20the%20growing%20process

Mallawaarachchi, V. (2017, July 8). Introduction to Genetic Algorithms — Including Example Code. Medium. https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3

Binary/Decimal/Hexadecimal Converter. (n.d.). Math Is Fun. Retrieved 1 May 2021, from https://www.mathsisfun.com/binary-decimal-hexadecimal-converter.html

Pawar, S. N., & Bichkar, R. S. (2015). Genetic algorithm with variable length chromosomes for network intrusion detection. International Journal of Automation and Computing, 12(3), 337–342. https://doi.org/10.1007/s11633-014-0870-x

Y, T. (2020, March 26). Guide to Lighting + Lamping: Color Temperature, Color Rendering and. . . YLighting Ideas. https://www.ylighting.com/blog/guide-to-lighting-lamping-color-temperature-color-rendering-and-lumens/

pH Scale Defined - What is pH? (n.d.). JAN/SAN CONSULTING. Retrieved 3 May 2021, from https://www.jansanconsulting.com/ph-scale.html

Mamdani and Sugeno Fuzzy Inference Systems - MATLAB & Simulink - MathWorks United Kingdom. (n.d.). MathWork. https://uk.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html

Mamdani and Sugeno Fuzzy Inference Systems - MATLAB & Simulink - MathWorks United Kingdom. (n.d.-b). MathWorks. Retrieved 3 May 2021, from https://uk.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html

Vieru, C. (2018, April 4). AWS IoT-Driven Precision Agriculture. Amazon Web Services. https://aws.amazon.com/blogs/iot/aws-iot-driven-precision-agriculture/

Negnevitsky, M. (2011). Artificial Intelligence: A Guide to Intelligent Systems (3rd Edition) (3rd ed.). Pearson Education Canada. https://www.pearson.com/uk/educators/higher-education-educators/program/Negnevitsky-Artificial-Intelligence-A-Guide-to-Intelligent-Systems-3rd-Edition/PGM995577.html

Masoum, M. A., & Fuchs, E. F. (2015). Optimal Placement and Sizing of Shunt Capacitor Banks in the Presence of Harmonics. Power Quality in Power Systems and Electrical Machines, 887–959. https://doi.org/10.1016/b978-0-12-800782-2.00010-5

Artificial Intelligence - Fuzzy Logic Systems - Tutorialspoint. (n.d.). Tutorials Point. https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_fuzzy_logic_systems.htm

P. N. Suganthan, P. N., Hansen, N., Liang, J. J., Deb, K., Chen, Y. P., Auger, A., & Tiwar, S. (2005, May). Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Centre De Mathematiques Appliquees. http://www.cmap.polytechnique.fr/~nikolaus.hansen/Tech-Report-May-30-05.pdf

How to Measure Light Intensity. (2020, May 27). BIOS Lighting. https://bioslighting.com/human-centric-lighting/how-to-measure-light-intensity/

Mao, L., Chen, Q., & Sun, J. (2020). Construction and Optimization of Fuzzy Rule-Based Classifier with a Swarm Intelligent Algorithm. Mathematical Problems in Engineering, 2020, 1–12. https://doi.org/10.1155/2020/9319364

Yang, X. (2020). Nature-Inspired Optimization Algorithms (2nd ed.). Academic Press. https://doi.org/10.1016/B978-0-12-821986-7.00015-9