DOSAN: Dynamic dataset generation from remote multiple streams for AI RNN based forecasting prediction

Ian Whittemore

Masters by Research

School of Computing

University of Buckingham

Supervisor

Professor Ihsan Lami

24/08/2021

Contents

Table of Figures

Table of Tables

Glossary of terms

Term	Meaning
ANN	Artificial Neural Networks are a model that resembles the human brain in terms of processing
RNN	Recurrent Neural Networks are a type of ANN that deals with time series along with sequential data
ML	Machine Learning is the understanding of getting computers to learn and resemble the human brain.
LSTM	Long Short-Term Memory is used typically with RNN for learning long-term order dependence in sequence.

Acknowledgement

I would like to use this opportunity to thank my supervisor – Prof. Ihsan Lami for his supervision, encouragement, and guidance during my MSc degree. My gratitude extends to Russell IPM for the funding opportunity to undertake my studies. I would also like to thank my friends and family who, without their support, would have not been possible.

Abstract

A typical AI analysis & prediction cycle includes the creation of datasets, each of which requires a dedicated AI-Model that needs to be trained and tested. This cycle is typically done manually and requires several iterations. This cycle becomes even more complex when repeated for different datasets from various databases of many data streams. DOSAN is the name chosen for the solution that uniformly automates this cycle, saving much valuable processing time and avoiding human errors.

This thesis documents the research and implementation that led to the development of this solution. DOSAN is a system that: (1) automates the dataset generation from various databases; and (2) performs on-the-go AI modelling, training, and analytics of these different datasets. Therefore, achieving a reliable, accessible, and fast processing AI-based analytics and prediction cycle. It can be used with any database collected for any application.

The literature surrounding the prediction for forecasting (using AI or traditional methods) has been studied thoroughly for applications ranging from power generation to stock exchange. All relevant publications that helped to shape the design DOSAN have been reviewed in this thesis.

The implementation of DOSAN is carried out in 3 modules. The first module generates a dataset from any number of databases automatically. It collects all the variables of the selected databases for clustering by anyone via a simple GUI and then pulls all relevant data into a single dataset. It can do this for any number of combinations of data so to support various predictions on them. The second module builds an AI model tailored for any given dataset, then performs training on this model before doing the analysis and predictions. It does this by taking the dataset and feeds it into an RNN LSTM model that will be available via a simple GUI to modify the conventional hyperparameters, if needed, to allow for optimal training of the model. After the model has been trained, then a variety of options can be invoked to check on the outcome of the model and give insight into how effective it will be. Finally, the third module performs the predications (analytics on the dataset) which are displayed in various forms to promote easy analysis for further research purposes.

Several database scenarios of various IoT sensor nodes networks from agricultural applications have been used to test DOSAN to ensure accuracy and efficiency. DOSAN forecasts take minutes to produce whereas other systems and methods can take weeks.

As well as DOSAN, this thesis includes the development of two Applications that has been used to collect and monitor data on the go:

- CROM - Collective Reporting App On Mobile devices. This App allows for manual logging of data via QR scanning or photo entry. The data that is inputted is stored securely in a cloud database when the device running the App is online. The stored data allows for reporting along with an overlayed google map to show the location of data. The accumulation of the data extends to RAW and DOSAN.

- RAW - Reporting Analysis Web application. This App extends the functionality of CROM and offers a superior analysis of the data, collected with CROM, for reporting. This is accomplished using data analytical graphs and comprehensive GUI and filters on both a mobile device and a PC.

DOSAN was developed over the 12 months for a Master by research degree. DOSAN is currently used by the sponsoring company of this research.

Chapter 1 Introduction To DOSAN

Artificial Intelligence (AI) processes typically involve manual pre-processing along with careful consideration of data and the importance of hyperparameters. This being the collection of data along with formatting the data ready for model training. Often this becomes a repeating cycle.

An example of such application is pest monitoring and infestation forecasting before the insects eat the crop. For such an application, the data from the monitoring traps distributed in the farm, in the region and what is happening in other farms all over the country becomes relevant and must be considered. This data can vary depending on the farm deployment of IoT networks and region of interest. Only when combined with other available data (regional, geographical, historical) and from other IoT networks, an accurate forecast can be determined. Therefore, it is important to combine all data resources in an automatic detection system to quickly establish an early warning that a pending insect infestation is inevitable, and that is when DOSAN is most valuable. Right now, farmers are using manual processes that will take weeks to achieve a definitive decision on spraying or not. Also, not all variables (or relevant data streams) are used to give a quick informative decision or accurate forecast. For example, delta traps are widely used in farms to manually collect insects trapped in them each week. But because the analysis of this data is currently manually done, the forecast is inaccurate and is a lengthy process [1]. In a nutshell, a lot of crops can be saved if enough monitoring data is collected and analysed in a quick turnaround AI system with data combination.

This project has attempted to generate a system that automatically combines different datasets and then analyses the data intelligently to detect insect infestation in a short timeframe.

We have used data from this agriculture application to prove DOSAN. We have developed trend prediction methods that identify patterns linked to insect infestation. This project utilises ML (Machine Learning) technology for this solution. This thesis proposes to combine variables from any different database (DB) automatically to generate new datasets. These new datasets, Weather, Crop and Insect, will then be utilised in the ML model for prediction alongside data analytics.

The final version of DOSAN is tested and shows it will reduce overall prediction time from typical 10 days to few minutes. This is due to it being fully automatic and user friendly so that non-technical personnel can also use it.

We started this work with a situation that there was no automatic data collection, therefore CROM (Collective Reporting On Mobile) was designed to make this process easier and more efficient. RAW (Reporting Analysis Web) was designed as a sub-system of CROM to give

farmers and database owners further insight into their data. Lastly is DOSAN, the heart of this research, to make the combination processing of datasets and model training along with hyperparameters fast and efficient. The build process for each module is detailed as follows:

1.1 CROM:

We utilised Flutter [2] as a framework with a UI library that allows for fast native mobile and web applications. It hosts the design of the UI, and so it is made simple by using GUI elements such as drop-down lists and buttons for easy input. To build the mobile application, Flutter is used along with Android Studio [3], which compiles all the necessary files in an easy to navigate format.

The backend for the mobile application was accomplished using Firebase [4] and APIs (being Google Maps [5] and Weather [6]). The use of these services is to allow for secure cloud storage from the data provided by the user as well as utilising provided information such as weather reports and Google maps. The APIs allow the processing of data entry, such as users location to gather weather and geographical LLA (longitude, latitude, and altitude) coordinates for further analysis detailed in chapter 3.1.

This module was the first to develop and took 3 months. CROM helped me understand the projects data types and learn new technologies including programming languages.

1.2 RAW:

RAW is built using Flutter as it too creates web applications similarly to mobile applications with only a few differences. These differences are the Integrated Development Environments (IDE's), which was built through Visual Studio Code [7]. Using Flutter for web applications allowed for more powerful UI packages to be used for data visualisations such as bar graphs and line graphs.

Similar to CROM, RAW utilises Firebase to access the datasets generated from CROM users. The link to Firebase allows for dynamic graph generation that updates whenever the user from CROM uploads new information. This meaning that data analysis is performed live for the user without any delays.

This was the second module development, taking 1 month, and took my learning to the DB arena and server queries while reinforcing my learning from CROM.

1.3 DOSAN:

DOSAN was developed in 3 steps, ensuring that each step is fully functional and has been tested thoroughly before commencing the next step.

- The first step to implement is a “merge algorithm” using Pandas [8] for the dataset generator that takes multiple datasets regardless of size. The algorithm takes the chosen dataset(s) and compiles a list of all columns that are used which is stored in an array. The array will be displayed in a dynamic button generation function in the graphical user interface (GUI) for easy selection of columns to use. The analyst can then, via the GUI, decide on any combination of these variables for studying the relationships between the data available. Once all columns of interest are highlighted by the analyst, then a new dataset is generated. This process can be repeated for all and any combinations of data information. For instance, a farmer can discover how humidity affects a specific insects infestation trend.

- The second step’s focus was on the AI automation implementation. This is based on ML with Recurrent Neural Networks (RNN) alongside Long Short-Term Memory (LSTM) from the newly compiled dataset. RNN are renowned for their ability to train effectively on time series data which is the reason for choosing this network technology. The LSTM will allow the encapsulation of data from large timelines for better accuracy. A unique aspect of the program is the user ability to modify hyperparameters of the model for better accuracy. As can be seen in chapter 2.1, an in-depth study of other possible Neural Networks has been carried out in the literature review to ensure that the best prediction is achieved. Within this study, there was testing done for both multivariate and univariate types of scenarios. Hyperparameters in Neural Networks play an important role as it determines the structure of the network. Testing of different hyperparameters was had to decide if the analyst should be given a GUI to change hyperparameters. This testing proved to be useful as hyperparameter tweaking for different datasets has monumental differences in model prediction.

- For the output of DOSAN, a variety of options is given for data analysis and information on the model that has been trained. This is to ensure that the data is useful and that the model is trained for optimal efficiency. Predictions and data analysis are displayed in graphical frameworks for easy understanding. This includes graphs such as scatterplots, histograms, bar charts, line graphs, dot graphs and scatter plots.

Testing each of these 3 modules is detailed in chapter 4. Suffice to say here that significant efforts (almost half of the development time) were spent on testing and ensuring that the modules work to requirements/spec. These tests ensure that a full range of functionalities is met for CROM, RAW and DOSAN to fulfil specifications are met. This was conducted using formal methods.

Results of DOSAN testing are centred around combining several scenarios of data from several datasets to ensure the function is stable. Multiple ML models were also tested from

the multiple datasets which were all successful. DOSAN allows for a great time saving to be had which entails faster deployment.

DOSAN module took 6 months to develop due to the extensive learning that was required. DOSAN greatly helped my perception of AI & ML along with new technologies and systems that granted me two additional certificates. I was also able to appreciate the user experience side of software during GUI production.

The rest of the thesis focus will be on the literature reviewed in chapter 2, the details of CROM, RAW and DOSAN implementation with tools used are detailed in chapter 3 and detail of the testing and results in chapter 4, and with conclusions in chapter 5 along with future proposals to make this system cloud-based, multiuser, and robust.

1.4 The Problem Statement for the sponsoring company & use case

Pests in the agriculture industry pose a large risk, not only for the loss of revenue from produce damage but can damage equipment and property [9]. Crop damage is often caused by insects which are responsible for two types of damage: 1st is direct damage where insects eat leaves and create holes in the produce, and the 2nd is indirect damage where insects eat very little of the crop but transfer bacterial diseases to the produce [10]. The damage caused equates to an average loss of 30 per cent each year globally, year on year since 1940 [11].

Furthermore, farmers spraying crops with pesticides in anticipation of such insects has been banned. In the EU farmers must prove there is an infestation before any spraying of chemicals can be carried out. This has necessitated monitoring and forecasting of insects in crop farms. There is evidence that some farms have installed some insect monitoring traps (e.g., delta trap and camera-based images) but this area is developing, and some farms are using IoT sensor networks to collect information about insects. An example of such a monitoring system is using Unmanned Aerial Vehicles (UAV's) that collects data relating to crop disease management along with field-level phenotyping [12].

In the industry, the current options to assist in infestation control or analysis are largely expensive and inaccessible to many small family-run farms. Only 52% of all farms in the US use smartphones and desktops to assist in farming practices [13].

An accessible and cost-effective analysis tool is a Delta Trap. This works by a sticky paper grid for someone to manually collect and count to gather how many insects have been caught. This is time-consuming and can be inaccurate due to human error in the counting process thus giving false analysis.

Another option for analysis is the use of UAV drones which use photo analysis to monitor crop health. It does accurately notify farmers when there is a change in the health of the crop, however, it does not provide warnings before health deteriorates and is very costly.

Russell IPM develops Integrated Pest Management (IPM) technology to assist in using fewer pesticides and creating pheromones that can be sprayed onto fresh produce to keep it pest-free. Russel IPM are looking to integrate further technology into their products so that there are records of insects and a way of generating models that can be used for future value predictions on infestation trends. The problem was that their data was either scarce, scattered or limited as they only needed specific fields of data previously that would take unnecessary time and effort to separate and go through.

Russell IPM is proud to sponsor this research so that this may be integrated into their technologies. The systems produced will be used by them to:

- CROM will be distributed to farmers to log insects that they observe.
- RAW will be used by managers of farms as well Russell management to identify the scene in the vicinity of any farm as well as geographical trends and further analysis of the data recorded by CROM.
- DOSAN is being used by the company to provide forecasts and help them gear their products to be ready if needed and assist in finding areas that are causing an infestation.

1.5 Summary of Objectives

This project is made of three systems that will make up the whole project. Two systems are for data collection and statistical reporting. The third system is for data combination and time series model generation along with analysis for both the model trained and the dataset used.

The objective(s) for the three systems are as follows:

Mobile Application - CROM:

- Secure account creation and login.
- Acquire data from users through the means of manual entry or scanned entry. Data is then stored in a cloud database that will be used for model training as well as reporting.
- Reporting at a low level to summarise recent entries both manual and scanned.
- Incorporate Google Maps API to visualise trap locations.

- Weather API to provide current location information.

Website Application - RAW:

- Secure account creation and login.
- Providing further analytics to the data provided by CROM through the means of a dedicated website.
- Graphical graphs to display data.
- Search for data from a given date or within a period.

Dynamic dataset generation with AI forecasting Application - DOSAN:

- Combine multiple datasets/databases into a singular dataset that is fed into an ML algorithm for forecasting along with analytics on the model and dataset used.
- Ability to download the model after training.
- Show trend analysis between user-selected dataset columns.

1.6 Project plan

This research proposal requires several objectives throughout the course of the studies. This entailed gathering requirements along with the literature research to establish a good foundation, assessing the requirements and literature to learn the new technologies that needed to build CROM and RAW, implementing CROM and RAW, learning algorithms and technologies utilised in DOSAN, implementing DOSAN along with sufficient testing, producing an article for publication and lastly the thesis of this MSc Research. The plan was for the first 3 months to be focused on literature reviews along with the development of CROM. This was followed by 2 months of developing RAW while continuing literature research. Concluding with DOSAN that took the majority of the research of 6 months. Due to COVID-19, 2 months were taken out due to personal matters however the thesis continued upon return along with working part-time.

Chapter 2 Literature review

Dozens of papers were reviewed to understand the various implementations previously done for time series forecasting. The following is a review of the publications that directly impacted the proposed solutions, appropriately sub-sectioned for clarity as shown in the fish diagram in Figure 1.

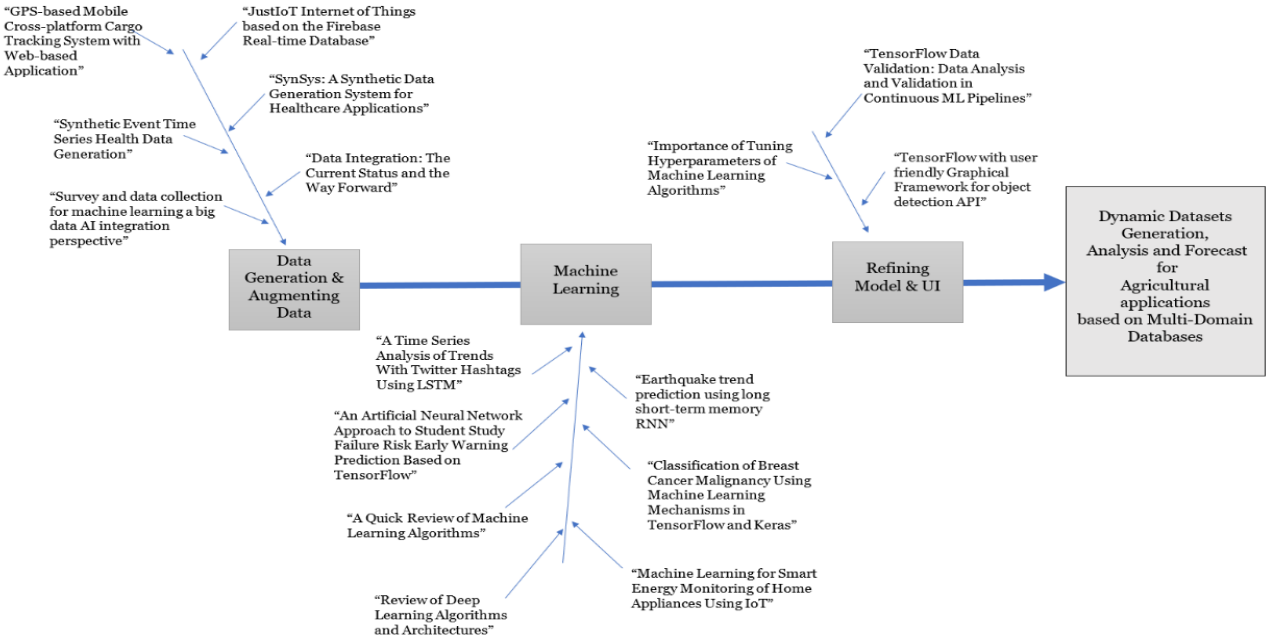


Figure 1 - Fishbone Analysis for Literature Review.

2.1 Data Generation & Augmenting

Cross-platform development is becoming increasingly used in the industry, this being due to cross-platforms using a single code base that works across various platforms to reduce expense and time. Tracking of cargo is not accessible to everyone due to this cost and time so this solves a problem for people who are looking to transport cargo but are not part of a large business. The authors of this paper wanted to illustrate this by building a cross-platform application with a web application for tracking cargo through Global Positioning System (GPS) [14].

The authors looked at several papers regarding GPS tracking, cross-platform development, and tracking methods. The system from the authors uses Flutter as the front-end development of the mobile application along with the backend as Firebase for the database. This works for Android and iOS. The web application uses Node.js as a back-end server that can work with various operating systems. This all provides services to transport companies and their clients to manage the transport of cargo in a particular country.

The authors used Flutter [15] to produce their mobile application along with Google Map API that is used to show cargo location. Flutter is the main technology for their system to develop mobile applications, this is a framework by Google built for cross-platform development. Following this is the use of Dart which is the object-oriented language for Flutter. The authors needed a Back-as-a-Service (BaaS) to control the real-time data that is required, this was done using Firebase, also backed by Google. To build the website, HTML coding language is used along with CSS (Cascading Style Sheet), which was used with HTML to build an appearance. The behaviour of the website is controlled using JavaScript to provide interactions. Lastly, Node.js was used to allow Javascript at the server-side, the use of Node.js was due to its high performance and high scalability.
The approach of cargo tracking is determined by the use of GPS. This application includes two sides, (1) the transport side and (2) the client side. The transport establishment side of the application is the use of a delivery truck with the driver using the GPS on their phone, thus transmitting the coordinates to the Firebase database, these coordinates are used with the Google Map API that has the location & address of the position. Once transmitted, these coordinates for the cargo may be accessed by the mobile application by clients or by using a web application where they could search for their cargo. The client-side is the use of the mobile app and web app. The mobile app allows clients to perform a function such as scheduling cargo for pickup, monitor their cargo status and finding the place of cargo utilizing the map. Submissions are then passed to the transport company for processing, once successful then customers will be able to gain access to the information and location of the cargo. The web app is similar to the mobile app however the difference is that there is the ability to modify fields such as city branches.
This proposed system from the authors with the methods demonstrated that Flutter is a tool to be utilised in the production of the systems of this research paper. Firebase too seems like a good, backed tool to use as a dataset however this paper did not talk in much detail.

This is looked at in further detail in this paper [16], the use of new technologies and new architecture is being used to create a new Internet of Things (IoT). This new scheme is meant to be safer, more robust, simpler to develop and maintain than standard IoT systems. The authors created this work to help the implementation and teaching of IoT in both educational and commercial sectors. For this example of their system, users can build a system to serve clients and operate a business.
After assessing tests and research, the authors created a new IoT system, JustIoT, which is split into four areas:

- The back-end - This uses Google Firebase for a real-time database.
- The front-end - Which is a SPA (Single Page Application) web monitoring application along with a mobile monitoring app.
- The controller for the connection between software-hardware.
- Lastly, an intelligence server that supports Message Queuing Telemetry Transport (MQTT) connection and condition control.

All of this enables users to set the rules for controls, remote monitoring, and control. The management web page has been built on Angular front-end technology that is connected to Firebase for a real-time database.
The authors look at multiple papers along with a variety of open-source integrated controllers such as Arduino [17], Raspberry PI [18] and Puppy Linux [19] to come up with a methodology of tackling multiple different controllers. For their research, Arduino is used. The build focus is primarily on the Firebase cloud system user-authentication and real-time database. Where controllers are directly linked to its database to read and write data. The authors chose Firebase as it offers access through iOS, Android, JavaScript SDK, REST API and Admin SDK. These are broken down into the following:

- iOS and Android SDK are designed for mobile app development.
- JavaScript SDK is intended for web applications.
- REST API is a common network service, it could be used as a general programming language.
- Admin SDK is used for the Node environment to carry out the JavaScript applications.

Firebase ensures that these links are encrypted. The other build focus is on information visualization and the management of JustIoT primarily from the browser and mobile phones. To build the mobile applications, the authors used Ionic SDK [20] which develops hybrid apps. The intelligence server in JustIoT has three functions: data transfer, conditional control rule engine, and machine learning. The supervision and management program for JustIoT is an Angular application. It is a static page that is easy to be hosted into a common cloud system. The primary functions of the monitoring and control system are user registration, system-associated information setup, condition control rule settings, data visualization, past data visualization and remote management. The intelligent server can be seen as a bridge between Firebase and the vulnerable controllers, which carries out the transfer of data and remote commands. The authors used a straightforward smart house IoT application with an Arduino controller. The smart home system includes three inputs: humidity, temperature, and brightness, it also features two outputs: living room light and bedroom mood light. When the Arduino controller is attached to the MQTT server, the authors can monitor the smart home system and gain remote access to its outputs with the JustIoT SPA supervision and management program. The author's program was a success as it was able to complete all objectives that it looked to achieve.
This paper also had many insights that could be taken away, however, one of the primary takeaways was the use of Firebase and the reason they used it in comparison to others.

Now with some of the front-end and back-end technology found, the next stage was to look at the use of creating synthetic data realistically. To accomplish this, papers were reviewed to assess the methodology to use.

Often existing methods for creating synthetic data are restricted to complexity and realism. The authors launch SynSys, an ML-based artificial data generation approach, to help improve these restrictions and create synthetic data for health care applications [21].
To build SynSys to produce synthetic time series data, a method of using sequences of Hidden Markov Models (HMMs) [22] and regression models is used, that are initially taught on real datasets. The HMM provides benefits such as a strong statistical foundation, the handling of inputs of any length and the use of pattern discovery. However, an HMM cannot show the dependencies between hidden states and needs many unstructured parameters. The authors test their synthetic data creation methods on a smart home dataset. The method of testing is by using the time series distance measurement as a starting point to discover how credible the synthetic data generated is compared with the real data. SynSys is found to produce more realistic data in terms of distance compared to a method of arbitrary data generation. This was done using semi-supervised learning to preserve the underlying patterns of the real data.
Past work was looked at by other papers who attempted this, which is the use of GANS [23] Which uses a discriminatory model to establish whether the created samples appear to be from the actual data distribution. The GAN model continues to attempt to improve the realism of the generated samples until the model cannot distinguish the real from synthetic data. However, the authors did not use GANS as they found it to be difficult to create synthetic data when there was no data to be had to learn from along with unstable training issues that occurred. So, their approach uses HMMs which they found to work well with the problem of modelling smart home data. While the authors found this to work for them, this type of implementation would not suit the purpose of this research as data needs to be dependent on one another along with a structured set of constraints. However, the idea of GAN models may prove useful.

While looking at synthetic data generation still, another paper was reviewed to accomplish the same task, however, this too was targeting health datasets. This seems to be a common trend when searching for literature. The authors wanted to look at creating synthetic medical data that maintains the privacy of patients while being a utility that could be used as a viable alternative to real medical data [24]. Often research is prejudiced towards those rare publicly anonymised medical datasets.
There is the inherent complexity of the actual data, such as that every patient visit is an event, this meaning that authors needed to transform the data through the means of using statistical summary. That characterize the events for a specific set of time intervals. After this, the authors train a GAN model to generate synthetic data. The authors tested their model by producing human sleep patterns from a publicly accessible dataset. Once generated, the authors evaluate the data produced by showing how near the univariate similarity between synthetic and real data is. The author's goal for this research is to construct an end-to-end system that produces artificial health data that captures relations of patient records including their covariates. Their system is aimed to not need domain knowledge.
To demonstrate their approach, the authors generated sleep patterns from the American Time Use Survey. Their tactic is to transform this event information into cross-sectional

data comprising of one record per subject. Originating in the dataset, the authors consider these events, such as the features, and attach them to the covariates of age, sex, day of the week and month of the year. Therefore, this dataset is transformed to cross-sectional data which is made up of 34 features per patient. As their main source of work, the authors applied HealthGAN [25] (a Wasserstein Generative Adversarial Network) to produce the synthetic dataset.
Results demonstrated that a synthetic average sleep per hour is similar to the real data. The generator used is also able to learn the covariates from the features in the dataset. This is seen as the covariate possibilities between the actual and synthetic data are tightly laid along with a diagonal graph that they generated. However, even though the synthetic data successfully captures the trends, it miscalculates the average sleep time for teenagers. Which they reckon is caused by the anomalies and high variances in sleep patterns for teenagers.

Data generation using GAN models will not be used, as this too was found to be the wrong fit for the research undergoing. GAN models primarily focus on image generation and training to generate text is very complex and can be inaccurate for certain covariates as discovered through the authors testing.

The use of ML for synthetic data generation seems to be a challenging task that is not deemed accurate for use. There is much research that can still be had in this area, but this was not the focus of this paper, other methods would need to be considered.
Scalable data integration is a challenge in the enterprise based on experiences at Tamr [26]. The authors want to emphasize the practical considerations when using machine learning to automate manual or rule-based methods for data integration. The reason for this is that many large companies are split into separate business units to ease responsiveness. This results in waiting for approval from all the business units which means it takes a long time to get something achieved. This leading to data silos, where such information is being stored with various granularity and schema, that is contradicting each other along with inconsistent details [27].
The authors use multiple examples which emphasize the technical problems for creating a deployable and useful data integration software that will tackle the data silos problem. Tamr uses supervised machine learning to combine large numbers of data sources rather than using rule-based approaches. The authors discuss that traditional solutions do not scale and that further research is needed to tackle this. They found that using Master Data Management (MDM) systems, such as Informatica [28] and IBM [29], can perform merges/purges using a rule-based system thus allowing specialists to stipulate rules for converting and categorizing input data records. However, prior to this step, typically an exercise of matching columns using a GUI in a semi-manual way is taken. Therefore, the authors say that MDM systems do not scale well. As an example, when there are more than 1000 data sets, each with around 500 columns. Then manually linking 500,000 columns is not feasible, and thus a scalable and more automated solution is necessary.
The authors state that one approach for scalability is the use of ML to automate the steps, however, this will require excessive engineering and knowledge, as well as that it is very

expensive to implement and time-consuming. Another reason why the authors are hesitant to use ML is that there are integration problems in enterprises. This means that there must be some explanation as to why the ML model a specific action. As an example, if a model is designed to be predictive to generate approvals for loans, then an explanation should be had if someone were denied a loan. If this explanation is not clear, then legal action would be inevitable, thus the adoption of these models is unlikely. Instead, to tackle silos, the authors state that around 15% of information is missing or incorrect in a typical repository. The way to fix this is to first look at missing values and outliers through the means of standard techniques and then perform data wrangling transformations. Another data clean-up strategy is to create clusters of records, to symbolize an identical entity using a deduplication tool, however, using deduplication to generate clusters is essentially inaccurate and requires a certain level of human action.

There is an area to be commercialised to support the work of data scientists. The reason for saying this is that the authors state that Merck [30] data scientists spend approximately 90% of their own time searching for information related to their task at hand, then performing data integration on the outcome. This means that there is a detection problem to identify information of interest.

To create an interesting discovery product, these systems need to involve relationships between objects, for example, Column A in Dataset 1 is associated with Column B in Dataset 2. For example, a data scientist who wants to know about the information that links with the effect of medication on mice will need to find data sets dealing with the topic, then this scientist would need to know the correlation between the various data sets. Requiring a considerably more elaborate catalogue system, such as the Aurum project [31], where correlations are discovered using syntactic and semantic features. However, another problem appears, which is that a data integration project is frequently operated by a single data scientist. Meaning the same data scientist has to be both the domain expert and the project manager which is not a sustainable option.

Another area open to the future of data is the ability of model reusability. This should be done easily using the results from the data integration project as training data for a second project and so forth. Transfer learning as well as the reimplementation of ML models hold great promise as stated by the authors.

This paper gave some great inspiration to the implementation of this project. This inspiration is the use of linking columns from various datasets to reduce the time taken for data scientists to compile datasets and find correlations without being a domain expert. Another inspiration from this paper is to use the ability of model reusability through downloading the model and use it in other applications for either additional training or general use. The last point taken from this paper is that the use of ML model generation for a dataset is not going to be had as this cannot be explained well and there needs to be a level of existing data to be had first with a hidden trend. However, synthetic data generation will still be needed but in another manner.

Machine Learning is one of the default platforms for analysing large data collected for any application. This does not however take away the difficulty encountered to obtain the data in the first place. IoT and sensor networks are ideal systems for collecting huge amounts of

data by placing sensor nodes at appropriate locations and stream/forward their data messages into a host/application database via wireless and networked technologies. There are two reasons why the collected data need to be relevant when used in intelligent predictive tools: the first reason is that ML is increasingly becoming more used which results in new applications that do not have sufficient label data (known identifiable raw data that has been processed); the second reason is that ML require a large amount of labelled data for it to automatically learn and establish correlations/features at several levels of abstraction for learning complex inputs and output. ML models consist of data acquisition, data labelling and improvement of existing data or models, the details of data collection are in Figure 2 [32] (steps in BLUE mean that they are inadequately contributed in accordance with the data management community).

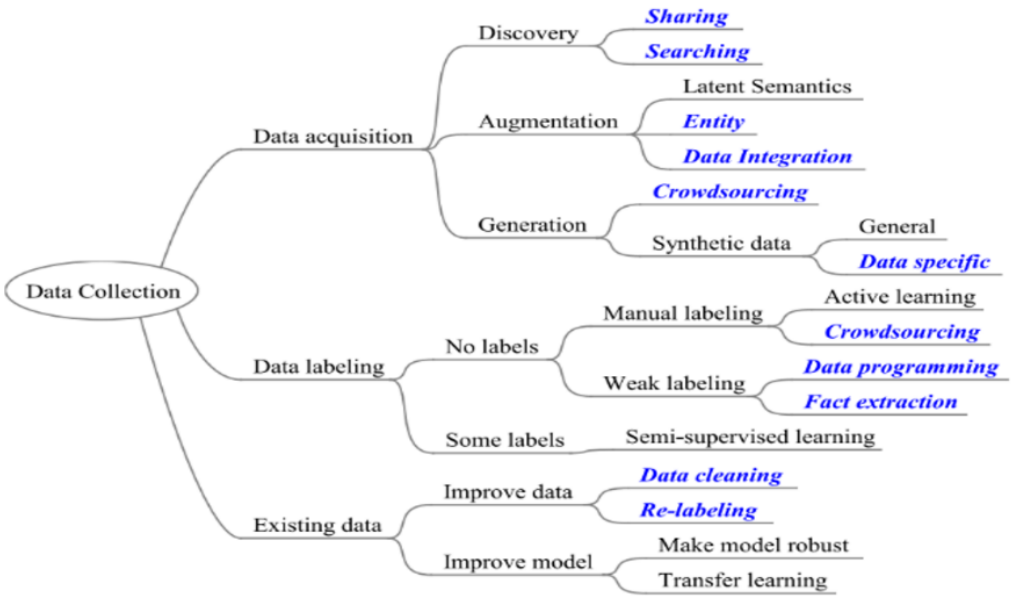


Figure 2 - Break down of the data collection methods.

Therefore, the challenge, for any developer of an ML model, is determining if the dataset is good for dynamic prediction or not. This is because it is hard to know if the collected data is useful until the model is trained and tested. Generally, there is a lack of good data that is accessible for small-budget projects. So, the authors of this paper review wanted to provide a specific guideline that documents how to source the best quality data possible in any circumstance for projects using machine learning algorithms. So, the authors have carried out a research landscape of the data operation types that have been carried out to bridge machine learning with data management. Also, the paper provides guidelines on which technique is best to use depending on the various research challenges on any data collected. To achieve this, the authors have identified three approaches for data collection: The first is sharing and searching for new data sets. The second is using data acquisition techniques to augment and generate data sets. The third is to improve an existing data set by cleaning up the data that already exists. A breakdown schematic of these methods is created in Figure 1

to allow scientists to distinguish which data collection method to use for any ML project [32] .
The authors studied 187 ML projects for data collection methods, the benefits of these methods and how effective these were. Illustrated in Figure 2, method (1) specifies 3 ways of data acquisition (discovery, augmentation, and generation). Discovery is broken down into two paths. The first path is sharing data with other developers/scientists using collaborative analysis which entails the community sharing datasets and then analysing the different versions which can be done through datahubs, which is a collection of data coming from multiple sources. Another option for sharing data is using the web through technology such as Google Fusion [33]. The second path is data searching, which can be further broken down into two paths. This involves searching through the company data lake where there is an abundance of legacy raw data stored that could be utilised for dataset collection, this can be optimised by using IBM products which creates data wrangling (transforming and mapping data from raw data into another format) for easy maintaining, storing, and filling.

The second step for data searching is also through the web but using systems such as Google Data Search (GOODS) [34] which catalogues the metadata of billions of datasets within google or using WebTables [35] that collects data using web crawlers and placing it in a tabular form.
If method (1) cannot be achieved through discovery, then another route is data augmentation which includes 3 paths (latent semantics, entity augmentation, and data integration). Latent semantics refers to the use of generating and embedding of representations, such as words, which can be used for Natural Language Processing (NLP) machine learning models. The way this is done is by vectoring words together which allows the model to generate surrounding words to form the text - the two possible models for developing this is to utilise Continuous Bag of Words (CBOW - used to estimate the likelihood of a word based on a given context) [36] and Skip-gram (Used to discover the most closely associated words for a given word or context). Another path to take is to use the technique of “entity augmentation” that is used when datasets are incomplete and require filling, this can be done using systems such as Octopus which gathers web search queries. Once gathered, the data is then clustered and from this, the ones with the most relevance are joined together to complete the dataset. Finally, data integration is the last step to be taken if there are a selection of different datasets that need to be combined to create one overall dataset. The authors found that machine learning toolkits often think that a training dataset is an individual data file so having smaller relational databases can impact overall performance as it does not need to sort through or navigate through multiple streams. Method (1) ‘s final thread is data generation, which includes crowdsourcing and synthetic data generation. Crowdsourcing can be split into gathering data and pre-processing data. Gathering data is obtained through programs such as CrowdFill [37] that allows workers to fill in partially filled in tables, another method of gathering data is the use of ALFRED [38] technique that utilises the crowd to generate data. Pre-processing the gathered data is useful to ensure it is optimal for machine learning purposes. Synthetic data collection is the alternative route to take place if there is no data to be had, systems

such as Syntactically Controlled Paraphrase Networks (SCPNs) [39] can generate text-based synthetic data. Method (2) is used when there is enough data but that needs data labelling. This summary is focusing on labelling techniques for regression to predict a continuous outcome variable and not classification. Method (2) consists of two branches for regression (Use existing labels and Crowd-based). The first branch (using existing labels) is self-labelling, this is where the model can generate labels for the data that is given. A framework that can do this is Co-training which uses two k-nearest neighbour regressors that label the unlabelled data. The following branch is Crowd-based techniques which are composed of further branches. The first branch is active learning where "interesting" unlabelled work is given to the crowd in the hopes that the work is more accurate and when utilised with regression and semi-supervised it creates a more robust model and less chance of overfitting (overtraining). The second branch is crowdsourcing, which is like active learning but is focused on producing the labels from workers who are not experts which means mistakes are had but can allow the model to learn from noisy data that helps create a more robust prediction model. The final path is a method (3) which is to be taken if there is already existing data that has two branches (Improve data and Improve model). To improve data, one must consider if to take data cleaning or re-labelling.

- Data cleaning is to reduce the noise of unnecessary data and ensure all data is in the correct format, tools such as HoloClean [40] can clean and follow quality rules to ensure that it is best used for ML.
- Re-labelling is not meant to increase the number of labels in the dataset but rather to improve the quality and be more selective.

The last branch is improving models to allow the model to be more robust against noise or biases. This can be done by removing the noisy labels and ensuring there is no overfitting and underfitting (too little training). The other method of improving a model is transfer learning which is reusing pre-trained models, thus utilising knowledge gained from a previous task to be used for new problems. A way to find similar models is using the TensorFlow hub. The authors found that even though these guidelines proved to be effective for datasets with machine learning integration, they also found that this can be subjective and implementing these can be difficult based on applications that can range from trivial to impossible.
These guidelines are easy to apply and follow in the development of the three systems. The inspiration from this paper is to consider combining semi-supervised labelling with active learning for effective model generation of DOSAN dynamic dataset generation application with ML analysis. Also, adapting the method of the augmentation so that users can create models on new datasets on the go, rather than keep using older models that can result in poorer predictions overtime with new data being fed into it while ensuring the model is being trained on a single dataset file for optimal results.

2.2 Machine Learning

The review focus is shifted into models and algorithms to use for application development. Several papers were chosen in this review to demonstrate the model DOSAN used and what insights can be had.

To start with building a machine learning model, one must choose and research all the right tools that will be needed along with the algorithms. A paper looking at the creation of an application in IoT, known as Cognitive IoT (CIoT) is looked at to create the decision-making process based upon historical data, automatically train, understand and solve problems with future issues [41]. The reason for this paper is because there is an issue globally of Inefficient energy use.
CIoT is using a new computing model called Cognitive computing that constitutes the convergence of IoT with machine learning. The authors looked at multiple papers in regard to HEMS (Home Energy Management Systems) [42] to help establish a foundation for their project. What the authors found is that the fundamental IoT architecture is a three-layer architecture that can be divided into 3 layers. These layers are:

- (1) The physical sensing layer, this layer utilizes a non-invasive current sensor that is attached to a Raspberry Pi 3A+ gateway.
- (2) The IoT middleware is the network layer. The use of Google Colab [43] is for model training. This is because the authors found it to be versatile and have the use of cloud solutions. This provides speed optimization by utilising GPU assets from Google servers to the developer's side, thus not needing internal powerful GPUs for training and testing. The dataset that is used is from UCI, which is an individual household electrical power consumption data set. This time-series dataset spans over 2 million minutes in 1-minute intervals which was further divided into a training and testing ratio of 90:10. The model generated using Google Colab is an RNN which includes an LSTM. The model was built with 75 LSTM cells firstly followed by 100 cells, the number of cells was discovered through trial and error. Finally, a dense layer was added along with an activation of mean squared error (MSE) to evaluate the fit of the model.
- (3) The application layer. In this paper, the Matplotlib [44] library was used to visualize the data. To have a function of a dynamic data stream, a Python program includes the FuncAnimation [45] method from within the matplotlib.animation module is being used. This allows, what was, a static graph to produce updates automatically on a regular basis and adding timestamps are placed on the plotted data.

Research findings from this paper found that there was a train score = 0.1798 with a test score = 0.1229. Showing that there is just a small deviation (closer to zero is better) from the original dataset. This paper greatly helped nudge the direction of this research into a more focused direction. The use of google colab for testing will be used as this will entail faster times to get started as no programs, software or languages are needed to be installed for use. The idea of RNN models seems promising but further research is needed before making a final decision.

There are several commonly used ML models to be used, but not all fit certain tasks. An evaluation of models is needed to gain a better understanding. A paper looking into the categorization of breast cancer malignancy at the same time using digital mammograms is had to assess some of these ML mechanisms [46]. The reason the authors chose this paper is that this area continues to be a challenging task in breast cancer diagnosis which is one of the world-leading cancers for women.
The authors investigated several machine learning mechanisms, these being Support Vector Machine (SVM), Logistic Regression, Decision Tree, Random Forest, and Deep Neural Network (DNN) which are used in TensorFlow and Keras. The authors also applied Python to estimate if a patient case is malignant or benign. This research is based upon the dataset presented by Breast Cancer Wisconsin, which has a set of 30 features. These papers showed multiple models with successful results.
The dataset that was used included around 600 patient cases that have suspicious-looking breast tissue areas. For each patient data contains an ID number and the breast cancer diagnostic results. The team created a simple methodology which is that the dataset will go into some data pre-processing which is then fed into each type of mechanism, this being the SVM, Logistic Regression, Decision Tree, and Random Forests, and DNN, from this an evaluation of performance is had. The data that was used was divided into a training and testing ratio of 80:20 along with the min-max normalization function. The testing was done on an Intel Core i5-6600 Central Processing Unit at 3.30 Gigahertz, and with 16 Gigabytes of memory. Along with this was an 11G graphic card used for training and testing. The software that was used was Microsoft Visual Studio and Python, with TensorFlow 1.11, Keras 2.2 and Sklearn 0.19 [47]. The results after testing showed the following in the format of sensitivity: specificity:

- SVM = 94%: 93%.
- Logistic Regression = 94%: 90%
- Decision Tree = 93%: 92%
- Random Forest = 91%: 95%
- Deep Neural Network = 98%: 91%

The authors believe that the DNN proved to be the most superior. These results have reduced the multiple paths that would have been needed to assess which mechanism to use. As it stands, the top two models for testing are the RNN and DNN models.

One of the most commonly used ML models is Artificial Neural Networks (ANN). This paper looks at using an ANN model to predict if a student will fail their course. The authors investigate this field as higher learning are faced with challenges to low course study completion and graduate degree completion rate. This failure of courses depends on various factors and currently outdated (still done by hand) prediction and analysis had to assess if a student needs assistance, this often being too late [48].

This study looks at implementing a system to predict early warning student study failure. This system will be based upon TensorFlow for building the model needed. This model will comprise four input variables: (1) Login times for the online education system, (2) number of times that were spent downloading study resources, (3) points that are earned through attendance, and, (4) Points earned through assignments. There is only one target variable which is the final course grade. After validation, the model showed prediction results close to the actual data.
Several papers were looked at in regard to why students fail along with methods such as association rules, decision tree, discriminant analysis, logistic regression, network analysis, and classification systems for predicting and classification. Overall, the authors felt and found that ANN was the model to use. The team used courses from their university to study 391 students as samples. Among these samples, 296 were randomly chosen for the network model training with the left over 95 to be used for validation. The target variable is using the hundred percentage points-based system with 60 points as the passing score. The authors quantified the students study failure risk employing a three-level risk classification system of red (R) for serious risk, yellow (Y) for moderate risk, and green (G) for a pass.
The ANN model included a Backward Propagation Network (BPN) [49] with one hidden layer that will be able to estimate an arbitrary nonlinear function. Therefore, this ANN included a three-tiered network including one input layer, one hidden layer and one output layer. To get the values set for the ANN model before final testing, the authors used the trial-and-error method to evaluate. The trials for the model were measured in relation to the value of RRMSE (relative root mean square error) along with the value of MARE (Mean Absolute Relative Error).
The problem that is seen through reading this paper is that this ANN model is primarily about constant value prediction, which the authors tried to solve utilizing the sigmoid activation function along with a linear activation function [50] for the output layer. This cannot guarantee long term dependencies though. Their model did perform to their objective and has some good prediction values. Therefore, ANN model testing will be had to evaluate for the task given for this research paper.

A short review of several machine learning algorithms that are often used was needed to establish which one is best used. The author aims to emphasize the merits and demerits of machine learning algorithms [51].
To create a review, the authors first needed to gather multiple papers of popular algorithms. The authors compiled a large summary of these algorithms that can be used to help give insight into what algorithm best suits certain tasks.
These algorithms are summarised into the following:

1. Gradient Descent - This is an iterative approach where the objective is to minimize a cost function. This works by means of computing the coefficients in each iteration by taking into account the negative of the derivative. As well as decreasing the coefficients in every step by a learning rate that is multiplied with a derivative so the smallest amount can be accomplished in just a few iterations. Therefore, the iterations are stopped when it converging to the minimum cost function. Gradient

Descent does have disadvantages, which are that whether the learning rate is too high (fast) then it's going to skip the true local minimum for optimisation of time. However, if the learning rate is too low (slow) then convergence may never be had due to it trying to find exactly the local minimum. The authors discuss having an evolving learning rate that slows down as the error starts to decrease is considered good practice.

2. Regression- this is a supervised learning method. Used mainly to model continuous variables and do forecasts. This requires a labelled dataset and an output variable value that will be decided by input variable values. Linear regression is viewed as the simplest form of regression to try to fit a straight hyperplane to the dataset. This is possible if the connection between the variables of the dataset is linear. The benefit of using regression is that it is easy to understand and can easily avoid overfitting employing regularization. The other benefit is that linear regression is helpful in understanding the data analytical process. The disadvantage though of linear regression is that this method should not be used for most useful applications as it generalizes real-world problems. The other drawback found is not helpful when dealing with non-linear relationships. Linear regression is often used in systems such as predictions and forecasting.
3. Multivariate Regression Analysis generally entails a single dependent variable that depends on several factors (A one-to-many relationship). As additional input variables are added to the dataset and model, it then creates a relationship among themselves. The advantage of using this technique is that it provides a deeper understanding of the connection between the independent variables and dependent variables that linear regression does not. The disadvantages, however, are that this is a complex technique and needs high knowledge of statistical techniques and modelling. This type of technique is often used for tasks such as forecasting at a deeper level, classification, and prediction.
4. Logistic Regression - Often associated with a classification problem. This technique gives the binomial outcome if an event will occur or not which is based on input variable values. Logistic Regression deals with the prediction of targeted variables in a categorical format. The advantages of using this are the simplicity of implementation, computational efficiency, and ease of regularization. The other benefit is that no scaling is required for input features. The disadvantages are, though, the inability to solve non-linear problems due to the decision surface being linear, this technique is more accessible to over-fitting and will perform well unless every independent variable is identified. Primarily used in applications for classification at an industrial scale.
5. Decision Tree- This technique is a supervised machine learning method that is designed to find a solution to category and regression problems. This can be accomplished by means of constantly dividing the information based on the specified parameters. Decisions are contained in the “leaves” while the data is divided into nodes. This means that the result is in the form of a yes or no format. The advantage

of using this method is that it is appropriate for regression alongside classification problems, there is ease in the interpretation and handling of categorical and quantitative values. The other benefit is the ability to fill data gaps in attributes by looking at the most plausible value. Decision Tree disadvantages are that it can be unstable along with the difficulty to control the size of the tree.

6. Support Vector Machines (SVM) are designed to deal with categorization and regression problems. This can be accomplished by looking at a set of objects belonging to different classes which later builds a decision plane to distinguish them and place the data into the correct classification group. The advantages of SVM are that it handles equally semi-structured and structured data. SVM is also less probable to overfit due to generalization. The other main advantage is the ability to scale up with superior dimensional data. The disadvantages of SVM are that the performance decreases with larger data sets due to an increase in the training time. SVM also will need a lot of data pre-processing as it does not function properly with noisy datasets. SVM are commonly used in applications to identify groups of classifications and determine values.
7. Naïve Bayes (NB) are based on conditional probability. It works using a probability table that updates through training data. This probability table works by basing feature values that need to be looked upon the class possibilities for forecasting a new observation. The advantages of using this technique are that implementation is easy, good performance is often seen, can be used with little training data, works with continuous and discrete data, used in binary and multi-class classification difficulties, and can make probabilistic forecasts. The disadvantages of this method are that models trained and closely tuned will outperform the NB model.
8. K Nearest Neighbour (KNN) represents a classification algorithm. Working through a dataset that has data points clustered in classes, an algorithm then attempts to classify the data point based on a classification problem. The advantages of this are that there is a quick calculation time and is versatile in terms of regression and classification. The disadvantages are that trying to classify unknown records is not easy. Data processing will be needed largely as noisy/irrelevant features will lead to poorer accuracy.
9. K Means Clustering Algorithm is frequently used for resolving clustering problems. This is part of unsupervised learning. The benefits of using this are that it is computing more effective than the use of hierarchical clustering techniques, this is because the use of globular clusters and small k results in tighter clusters. This algorithm is relatively easy to implement along with ease of interpretation of cluster results. The disadvantages are that the actual K value prediction is difficult to be had, along with performance struggles when clusters become globular. In regard to performance, this reduces when there's a difference between the size and density of the clusters in the input data.

Multiple algorithms were looked at and studied which allowed for an easy-to-follow summary of algorithms along with their advantages and disadvantages. From this study, the use of Multivariate Regression Analysis would best suit the needs of ML prediction and training.

One of the papers have helped assess the viability of using LSTM / RNN models in a prediction of gaming trends; so, it needs precise consideration for user's frequency, number of users, types of games, and connection quality to help the marketing and launch of gaming products for effective market penetration. The authors wanted to be able to predict upcoming cultural trends based on user-generated data such as tweets and how other users interact with this. They focused on twitter related to gaming to predict future gaming trends while solving the vanishing gradient problem of RNN's. The authors implemented an RNN model to uniquely identify gaming community trends that helped Twitter influence gaming marketing strategies. This has resulted in a clearer understanding of the importance of time series and the concealed trends that are had in the gaming community [52].
The team compared similar RNN's and took inspiration from this to assist in the generation of their model. After the team looked at four literature papers on ML prediction models, they were able to decide on the steps going forward for their application. The application was built using Python with NumPy, Matplotlib, TensorFlow 2.0 and Tweepy API to get the stream of data from Twitter and keep it in a CSV file.
Data evaluation was needed before application build to understand the behaviour and features of data. Time series data is compounded with base level (the mean value of time-series data), trend (increments or decrements slopes of the change of base-level over time) and seasonality (identified when a pattern is repeated). Time series datasets require both trend and seasonality. Data was collected from the 4th of October and concluded on the 28th of October as this was needed to acquire the training data for the model, the data includes all the ups and downs of a trend which should not be altered as it affects the outcome - this data will be smoothed as noisy (corrupted or distorted) data can also affect the outcome. The authors used Pandas to help with data indexing, training, and testing split for model training. The split included 192 rows of data which included 176 rows for training and the rest for testing. Data were normalized between 0 and 1 for enhanced learning speed by using the min-max scalar function as well as the batch function which only used 1 batch due to how small the dataset is.
The final RNN model consisted of 1 input, with 16-time steps for each batch due to the size of data. There were 200 neurons per layer which were done by trial and error. The learning rate used for the model was 0.001 and the iterations for training was 6000. The LSTM portion of the model included a mean squared loss function which is used to see that if the model predicted a value that is too high from the actual value, if this occurs then the Adam optimizer function assists in smoothing the values. The authors incorporated the Mean Absolute Error (MAE) which finds the distinction between the actual and predicted values and they found that their model obtained a mean absolute error of 20. 4935. This model was used to predict 2 days into the future which they felt were successful pre-dictions.
This is significant as it showed some great insight into building an RNN LSTM model and with behind-the-scenes numbers of the model which is hard to come by in such detail which

allowed the utilisation of their insights into the DOSAN model, such as the loss and optimizer functions for the LSTM and the min-max scalar function for learning rate.

This next paper helped reinforce the idea of using an RNN LSTM model and gave further insight into technical methods for the application build and how to further improve the opportunities that RNN LSTM models have to offer. The premise of the research is to attempt to forecast earthquakes by using an RNN LSTM model while smoothing out the bias errors that occur with scattered data. The authors want to compare these results with another Feedforward Neural Network model based on the same data to evaluate which is the better model to use. The use of this application will assist in determining if there will be an earthquake, what magnitude and where.
The authors created an RNN LSTM model that is used to forecast earthquakes and identify trends surrounding these as well as finding that an RNN LSTM model is more superior to an FNN (Feedforward Neural Network) model. To do this they compared how each model predicted given a set of data that had outliers in. The FNN model focused on the outliers and the predictions were skewed towards these. The RNN LSTM model was able to look at all the data and produce a less biased prediction [53].
The team researched twenty-one papers regarding prediction and using ML which gave them the inspiration for their models. The FNN model was created with 14 past earthquakes as input with the following earthquake being the aim. This model utilizes two hidden layers with 20 nodes in the first layer and 60 nodes in the second layer with each node having the sigmoid function as their activation. This particular model was trained for 1000 epochs. The next model was the RNN LSTM model, the reason for the LSTM was to remove the issue of vanishing gradients. The LSTM model was built using two hidden layers with 40 hidden units in each, there is also a dropout layer between the two layers for regularization, the model also included the loss activation of Root Mean Square (RMS) – similar to the MAE function but more useful for undesirable errors. The learning rate is 7 but decreases when the RMS loss is not improving after 10 epochs. The total amount of epochs used for the LSTM model was 1600 and was decided after trial and error. The two models were compared to determine which was the superior model for time series data.
The results were that the FNN was not able to correctly capture the different types of attributes that fit into the given data which made the results biased to certain areas, thus showing a weak trend forecast. The LSTM performed greater than the FNN model by 59% and was able to take into consideration all the attributes that were presented as input and therefore were not as skewed. These results have made it clear to use LSTM alongside the RNN as it will hold all data that is given with equal weight. A further impression was the use of the RMS function to evaluate the overall model if there is too much training being had or too little.

2.3 Refining Model & UI

With the model and techniques found, the next stage was determining how to improve on the model and decide what hyperparameters should be changed and how this could affect the model.

While deep learning has had much success in recent years, there are still factors that take place that make long training times. Setting hyper-parameters still is seen as a mystery that often is seen as needing years of experience to obtain. This study looks at various efficient methods to set up the hyper-parameters that should decrease the training time and improve performance [54].
This paper bases their approaches by looking at well-known concepts for the balance between underfitting versus overfitting. In this paper experiments discussed demonstrate that the learning rate, momentum, and regularization are closely related. This paper looks at a review of underfitting and overfitting, learning rates, batch size, cyclical momentum and weight decay.
Test and validation loss are seen to be a good indicator of the model's convergence. Achieving the balance between these is often seen as difficult in DNN models. This is why tuning hyperparameters should be taken with care and often require multiple tests to get a perfect fit. Some of these hyperparameters include the learning rates. Learning rates are seen to be the amount of regularization that is needed to be balanced for each dataset, as stated by the authors. The authors also found that decreasing means of regularization along with regularizing enormous learning rates should ensure more efficient training. Batch size takeaway is that a developer should try to get the best performance while keeping the need for computational time to a minimum. Batch sizes must be looked at in connection with the time it takes to execute the training of the model. Rather increasing the number of epochs for training ought to be had to enhance the performance while reducing the time taken. The authors discuss that momentum and learning rate are closely connected and that optimal learning rate is dependent on the momentum, however, momentum is conditional on the learning rate. The authors found that the most optimal training method is to combine an increasing cyclical learning rate where an initial small learning rate allows for convergence to begin. A balance is needed though where a large value for momentum can cause inadequate training results which can be seen early in the training. Lastly is weight decay, which is found that through experimenting with the maximum learning rates (once found) then this will be a straightforward solution and reduce training time.
In summary, the authors created a recipe for finding a good set of hyperparameters. This helped in evaluating the hyperparameters that will be used in the RNN model along with a methodology on how to assess training.

As some further data validation, another paper was chosen to give insight into this field. It is known that ML research is aimed at improving the accuracy and effectiveness of the training algorithms; however, this gives far less attention to the problem of understanding, validating, and monitoring. This is why this paper is chosen as it looks like creating a validation methodology to solve this [55].
This paper aims to approve a data-centric approach to ML that will look at the information being a high priority. The authors used a demonstration to showcase their TensorFlow Data

Validation (TFDV), which is a, as they state, scalable data analysis with a validation system developed at Google for ML. TFDV is aimed to be deployed as part of TFX [56], which is an end-to-end machine learning platform.
The authors of this paper looked at various papers and systems to assist in their research, this being focused on data cleaning literature that includes routinely detecting and correcting errors in the data. The authors felt that these works were limited to the ML scenarios that they were targeting. This was for the following reasons: (1) Data was coming from systems where updates were not feasible, (2) cleaning needed to be applied all the time for the training and serving path, and (3) Errors relating to data were often due to code were the only way to fix this was to patch the data rather remove the bug.
Once the authors found enough research, they started their work. This work included building a Data Analyzer which gets a set of statistics generators to compute statistical data that are necessary for analysing. They achieved this with Apache Beam [57] which defines and processes its data pipelines.
After this was the Data validation section, which is used to check the characteristics of the data as stipulated beforehand. These are the limitations linked with each feature that follow some fundamental properties such as type, domain, and valency. Lastly, is the Data Visualizer. TFDV creates a visualization for the statistics, schema and anomalies that are discovered. This was built using the Facets library to envision the statistics such as the mean, standard deviation, max value, and min values. Not only these but also graphs for data analysis.
This paper showed some methods of creating validation for data. However, the use of Pandas and Matplotlib will be had to create these validations as these are more commonly used and have more support than other methods.

In the literature, there is a lot of debate about whether hyperparameters tuning has benefits for ML models. This paper studies whether it is better to tune or not tune hyperparameters, and how this affects the model's performance. The authors have evaluated the most popular algorithms used within ML. They compared the results of each of the models using a given dataset, and whether the models used default or tuned hyperparameters. After evaluating fifteen papers, they discovered that the untuned hyperparameters gave better results than the tuned hyperparameters for each model. This means that scientists do not need to tune their hyperparameters to get good results and can simply use default values [58].
The application for the ML models and testing were created using Python on the Azure Cloud Computing [59] platform. The hyperparameters are tuned using a random search strategy and use a cross-validation procedure. The next step was using a two-sided non-inferiority test to compare the difference/similarities of hyperparameters (that are only interested in the risk in a fixed condition if the risk is higher than the risk is in a non-fixed condition). The two algorithms for hyperparameter testing are Support Vector Machine and Random Forest. The authors used default values for the models and applied them to 59 datasets from OpenML [60], which were of similar features and size to avoid biases. The models were compared simultaneously for the final evaluation of hyperparameters tuning.
In conclusion, this paper showed that leaving hyperparameters at default values makes no difference compared to using tuned hyperparameters. This was using two algorithms that

will not be used in DOSAN but have some similarities in terms of the idea of hyperparameter tuning. Further tests on models using DOSAN will be carried out to get a full understanding and comparison of each DNN (Deep Neural Network) model that is generated and change each hyperparameter to evaluate which is the best hyperparameter to change by the user if any. This methodology will be followed by tested using the two-sided test.

A Graphical User Interface (GUI) is needed for the frontend research and the following paper had many similarities in the field of tuning hyperparameters and model generation on data. The authors wanted to create an application so any user can deploy machine learning models without any coding while assisting developers.
They did this through a TensorFlow API (Application Programming Interface) that creates a GUI for the implementation of data pre-processing, training, and assessment in the client-side for im-age-based data. In addition, hyperparameter settings, real-time observations of the training process, object visualization of test images, and metric assessments of the test data are created. To create the TensorFlow API, the authors needed a good understanding of what typical end-user inputs to generate a CNN (Convolutional Neural Network) model. They also created a UI that allowed non-specialists to input data correctly which allowed the model to be trained without any additional coding needed. They were able to create a TensorFlow API, however with some knowledge or previous experience with Command Line Interface (CLI) to deploy the application [61].
All in all, twenty-four papers were reviewed before their application was built to give this team a thorough insight into their work. The backend application was built using TensorFlow 1.14, Python 3.5 and Anaconda while the GUI was built using Java Swing components. The authors used two servers which are the server-side which provides a deep understanding of operations as well as a client-side server that offers the interaction, and the GUI displays. To deploy the application a user will need to utilise the CLI to gain access and ensure that the connection to the servers has been made. This GUI does data pre-processing of the images by allowing the users to manually enter the labels of the images. After the training of the model is complete then a command line prompt with data is shown for the evaluation of the model which gives information about the loss and training of the model which determines if the model overfits (over trains) or underfits (too little training).
The results were promising as it showed a GUI that allows users of all backgrounds to create and train a model. The observations that were seen showed that some training is needed to use the software as it can be complicated. The application does not use the Graphical Processing Unit (GPU) for help in training and this is something that can greatly increase training speed. Due to this being a server application it means the server needs to be at optimal performance for everyone to use and access. Another issue is data centralization as this means sensitive information is uploaded into a server and while this can be made secure it could have large risks. This paper was significant to future work as there are ways to improve and remove these issues and challenges in future applications such as removing the CLI for deployment and display of evaluation.

Based on the research, this reinforces the need for three-level of systems for information abstraction. (1), a module dealing with user input along with low-level reporting, (2), a module to present high-level reporting and analysis and (3), combining multiple various datasets into one for ML processing with forecasting and analysis.

The literature reviews have assisted in compiling a summary of findings and a good footing to create DOSAN. The heart of DOSAN lies within ML, there are four types of commonly used neural networks [62]: Artificial Neural Network (ANN); Convolution Neural Network (CNN); Deep Neural Network (DNN) and Recurrent Neural Network (RNN). A DNN and an ANN are almost identical except for the difference that an ANN is only using one hidden layer [63] and if there is more than one layer then it becomes deep and therefore a DNN. Table 1 shows that CNN will not suit the purposes of this project due to its focus on image data rather than text. A further study between the ANN/DNN and RNN was needed to fully determine which model is best.

	ANN / DNN	**RNN**	**CNN**
Data typically used for	Tabular, Image, and Text Data	Audio, Sequence, Text, and Tabular Data	Image and Video Data
Length of Input	Fixed	Not Fixed	Fixed
Recurrent (feed their output back into their inputs recursively)	No	Yes	No
Spatial features	No	No	Yes
Effectiveness	Suggested to have less performance than RNN and CNN	Less feature compatibility compared to CNN	Suggested to have more performance than an RNN and ANN
Types of Applications	Forecasting, Recognition, Image Processing, Natural Language Processing, Pattern Recognition	Forecasting, Time Series anomaly detection, Recognition, Natural Language Processing, Pattern Recognition	Facial Recognition, Image Classification, Natural Language Processing, Document Analysis,
Advantages	Able to learn nonlinear functions, work with missing knowledge, can generalise, able to learn hidden relationships	Robust to noise, remember all information, great for time series prediction, inputs of any length, weights can be shared	Learns filters automatically, captures spatial features, parameter sharing, fast learning, robust to noise,
Disadvantages	Forgetful, not exact, tendance to overfit, processing burden,	Processing is slow, difficult to train, using activations such	Cannot learn temporal dependence, large

	needs a large amount of quality data, loses spatial features	as Tanh or Relu cause problems.	training data, Classification of Images with different Positions.

Table 1 - Model Comparison.

Deciding between ANN vs RNN resulted in multiple tests for comparison. These tests included multiple testing of the ANN to decide which ANN model is best, similar testing is had with the RNN. Once all testing was completed, then the ANN and RNN are compared to evaluate which one is better used. These tests showed, as displayed in Table 2, that the best ANN model had an accuracy of 0.9693 and loss of 0.0585, which is very good in terms of model generation but was found the model did not handle long term variables well and gave a sense of "forgetting" the initial data. The RNN on the other hand had an accuracy of 1 and a loss of 0.001, this showing better results than the ANN as well as being able to retain the initial data. For this reason, along with what was found in the literature review, the RNN will be the base model to be used.

Model	Layers	Epoch	Batch	LSTM	Accuracy	Loss	Val_acc	Val_loss
ANN 32 Neurons	3	600	256	0	0.9693	0.0585	0.9693	0.0585
RNN	2	400	128	25	1.0000	0.0011	0.9993	0.0011

Table 2 - ANN VS RNN.

From the literature and own testing of the RNN, the following parameters will be used in the final RNN generation for testing of the used dataset in chapter 4.3:

- Seed – Used to achieve reproducibility in ML by using the same starting point in a sequence to assist in reinforced learning [64].
- Batch – Number of samples the model works through before it updates the internal parameters. This acting as an iteration over samples while making predictions [65].
- Internal Evaluation – Process of estimating the performance of a model by training & evaluating the model multiple times using the same method [66].
- Step – Used as a forward-backwards evaluation of one batch. This is also used as the learning rate which determines how much change in the model is needed to the estimated error when the model weights are updated [67]. In other words, the response of replacing concepts the model must learn with new ones.
- Epoch – The training the model has with all training data for one cycle [68]. An epoch is made up of either one or more batches.
- LSTM – Used to learn order dependence for sequence prediction problems [69].

- Validation Steps – Used on the testing data rather than the training data to assess the model and ensure that there is no overfitting. This also tunes the parameters to classify the output [70].

Lastly, the RNN model will be semi-supervised. This decision was made because unsupervised models must work on their own to locate information, typically dealing with unlabelled data [71]. While unsupervised allows for more complicated processing tasks in comparison with supervised learning it can be erratic compared with other natural learning deep & reinforcement learning methods. Whereas supervised learning trains the model using well-labelled data, as discussed in the literature review, this meaning that data has been tagged with correct the answer. Further to this, supervised models help predict outcomes for unforeseen data as algorithms are trained using labelled data. This is highly accurate & trustworthy with the one drawback that classifying big data can be a challenge.

Chapter 3 Implementation

Requirement analysis highlights the expectations of users using the product. This analysis includes functional and non-functional requirements for the stakeholders. These stakeholders being:

1) Farmers will be using the application for input and output to enable them to get a better understanding of their farm life cycle as well as infestation predictions.
2) Russell IPM will use the information to assist in their pest control solution.

Functional requirements are services mandatory for the module to provide [72]. These requirements clearly label what should be implemented in the end module. These functional requirements can be viewed in Table 3.

No.	***Functional Requirements***
1	User should be able to login & sign up
2	Reports based on user's selection and choice should be displayed
3	The user's input is captured and saved in the cloud
4	TensorFlow used for RNN LSTM production
5	Use of Firebase for cloud storage and retrieval services
6	Flutter utilised for the build of Mobile & Web application
7	Multiple datasets should be able to compile into a singular dataset utilising a foreign key
8	Users should be able to view predictions with data analysis
9	Mobile applications should be downloadable from the Google Play store
10	Web Applications should be accessible through a dedicated web

Table 3 - Functional Requirements of DOSAN.

Non-Functional requirements are focused on the module's attributes [73]. These requirements serve as a control to ensure the module is usable and effective. Displayed in Table 4 are the identified non-functional requirements.

No.	*Non-functional Requirements*
1	User input is stored in cloud database within 30 seconds
2	Hyperparameters can be modified for slightly better performance
3	Training of RNN LSTM model should take no more than 2 hours
4	Google Map API incorporated for a different view of reports
5	Buttons and drop-down lists should be used for easy input and navigation
6	Colour palette for consistency
7	Implementation of Graphical Processing Unit (GPU) should be utilised if available for training model
8	Errors should be handled efficiently with correct messages to inform users without closing the entire application
9	Weather API implementation for further insight into
10	Robust to multiple processes/ entries simultaneously
11	Users' details are securely stored in correspondence to the General Data Protection Regulation (GDPR)

Table 4 - Non-functional Requirements of DOSAN.

3.1.1 Equipment & Software

A desktop and a laptop were needed to build the systems. While the desktop will be used primarily for building the systems, the laptop will be used for further testing and to ensure

compatibility. The specifications of the desktop and laptop are significantly different, as displayed in Table 5, with the desktop being the more powerful and the laptop being only subpar.

Components	Desktop	Laptop
Central Processing Unit (CPU)	AMD Ryzen 7 3700x at 3900 Megahertz with 8 Cores	Intel Core i5 7300HQ at 2500 Megahertz with 4 Cores
Graphical Processing Unit (GPU)	NVIDIA GeForce RTX 2080ti	NVIDIA GeForce GTX 1050
Motherboard	ASUS ROG STRIX X570-E	LENOVO Provence-5R1
Random Access Memory (RAM)	Corsair 64 Gigabytes of 3600 Megahertz	Samsung 16 Gigabytes of 1200 Megahertz
Storage	4 Terabytes of Non-Volatile Memory express (NVMe)	2 Terabytes of Solid-State Drive (SSD)

Table 5 - Dekstop & Laptop Specifications.

Four Samsung Android phones with different Operating Systems (OS) were used for testing and development, further specifications are displayed in Table 6.

Components	S6	S8	S10	S20+
Operating System	Android 5	Android 7	Android 9	Android 11
Chipset	Exynos 7 Octa	Qualcomm Snapdragon 835	Qualcomm Snapdragon 855	Qualcomm Snapdragon 865
CPU Frequency	2.1 GHz	2.45 GHz	2.84 GHz	2.84 GHz
GPU	Mali-T760MP 8	Mali-G71 MP20	Mali-G76 MP12	Mali-G77 MP11
RAM	3 GB at 1552 MHz	4 GB at 1866 MHz	8 GB at 2133 MHz	12 GB at 2750 MHz

Table 6 - Mobile Devices Specifications.

Taking the knowledge gained from the literature reviews in section 2.1, there will be several software, languages and methods needed to build the three systems. The following is what is necessary:

- Flutter – This is an open-source set of tools created by Google that enable the creation of intuitive and aesthetic applications that can run on Android, iOS, Web, and desktop [74]. Flutter is faster than React Native as it avoids the need to create a connection to interact with native components along with a hot reload for compilations of updates almost simultaneously [75]. Flutter utilises the programming language Dart [76], which was created by Google to leverage C-based languages which is why there is a "mixture" of elements of other languages amongst Dart.

- Android Studio – An Integrated Development Environment (IDE) for Android applications [77]. This IDE uses a Gradle based build system which is an automation tool that is flexible to build nearly any type of software as well as automatic download of dependencies and repositories for easy implementation [78]. The other key factors of Android Studio are that it comes with an emulator and already compiled files necessary for compilation. This software allows for the final build to be in the Android Package Kit format needed for the Google Play Store.

- Firebase – This is another toolset designed by Google which is a Backend-as-a-Service to build, improve and grow apps. Firebase handles and manages all databases with no SQL and gathers the hardware necessary, all the developer needs to do is initiate the API [79]. Firebase is not only used for cloud database management but also for cloud hosting, authentication and other cloud functions that are not being utilised for this research.

- TensorFlow – This is an open-source library that allows for creating ML applications. This library was built to utilise the use of GPUs to assist in the production of ML to save on time and processing power [80]. TensorFlow works by building dataflow graphs/structures on how data will move through by looking at the inputs and converting them into a multi-dimensional array. This meaning that input will go through multiple operations before it comes out at the other end as an output [81]. Python and C++ can be used to call upon this library.

- Pandas – This is an open-source Python package primarily used for data analysis and ML activities. Pandas was built on top of the package NumPy [82] that is used for support on multi-dimensional arrays, the reason for the use of ML [83]. Pandas was created to save time on repetitive tasks when working with data by including a vast library of functions in a one-word call function.

- Matplotlib – Created as a Python package to assist in the creation of data visualisations. Matplotlib make it simple by allowing developers to simply input data

into a function for visualisation [84]. Mostly used in conjunction with Pandas as it relates to data analysis.

- Python – Widely used for data analytics, ML, UI, and general applications. This is an object-oriented, high-level language for development [85].

- PyQt5 – This is mostly used as an open-source GUI module that bridges Python with the C++ UI framework [86]. PyQt5 comes with a Designer tool that can design interfaces and convert them into Python code enabling faster development as widgets can be easily implemented.

3.1.2 UML Use Case Diagrams

Use case diagrams show the relationship between the user and the system. It shows what is to be expected and the path the user takes to achieve a task from their perspective. Three use case diagrams are created to represent each module.

1. Module 1 (mobile application - CROM) use case diagram demonstrates the level of collaboration between the user and the mobile application. Along with the overall interaction, there is another factor which, in this case, is the database to show the connection between certain use cases.
2. Module 2 (web application - RAW) demonstrates a high level of interaction for analysis and reporting based upon the data entered from the mobile application.
3. Module 3 (ML & dataset generation - DOSAN) visually shows the connection and processes required for a user to train an ML model along with their own dataset generation. In the end, the user will be given insight on analysis for the model and the dataset.

Three use case diagrams, displayed in figure 3, figure 4 and figure 5, are designed to illustrate the user's perspective.

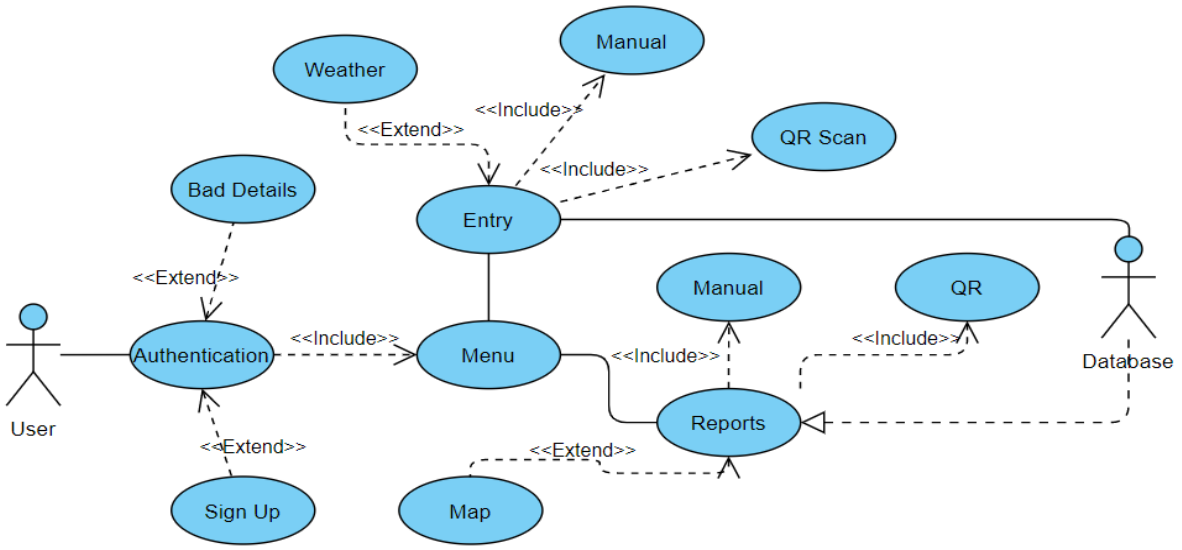


Figure 3 - Mobile application use case.

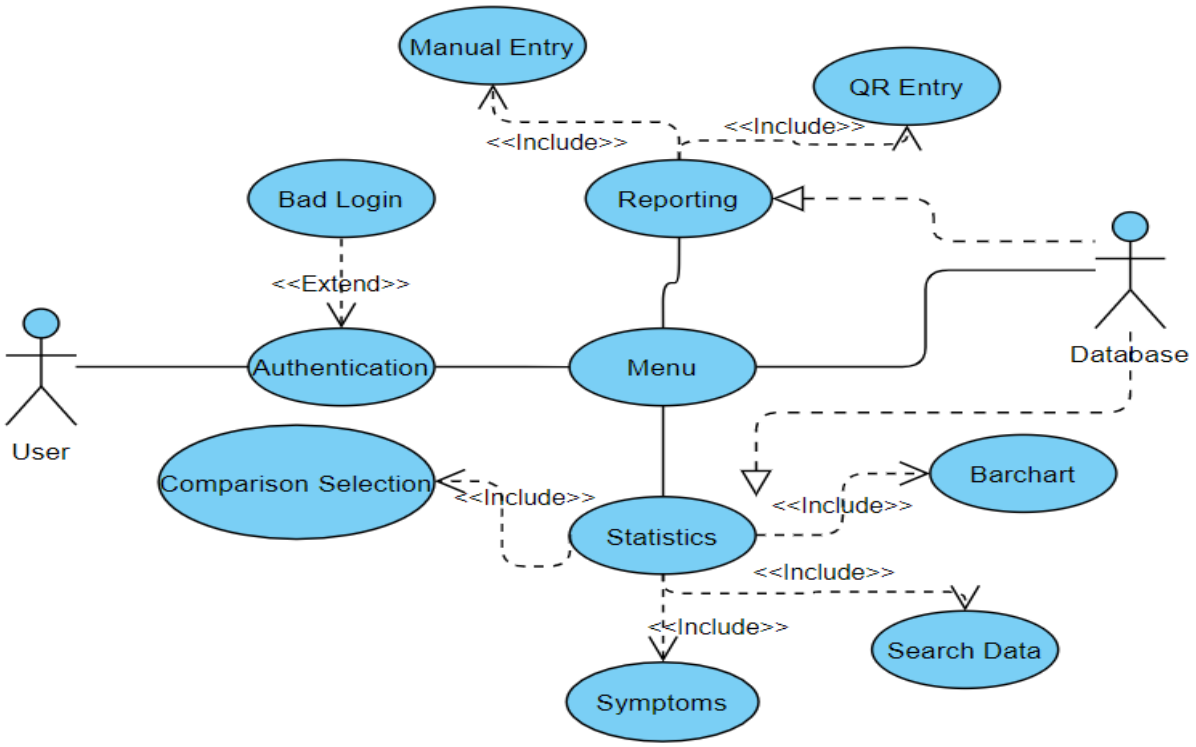


Figure 4 - Website application use case.

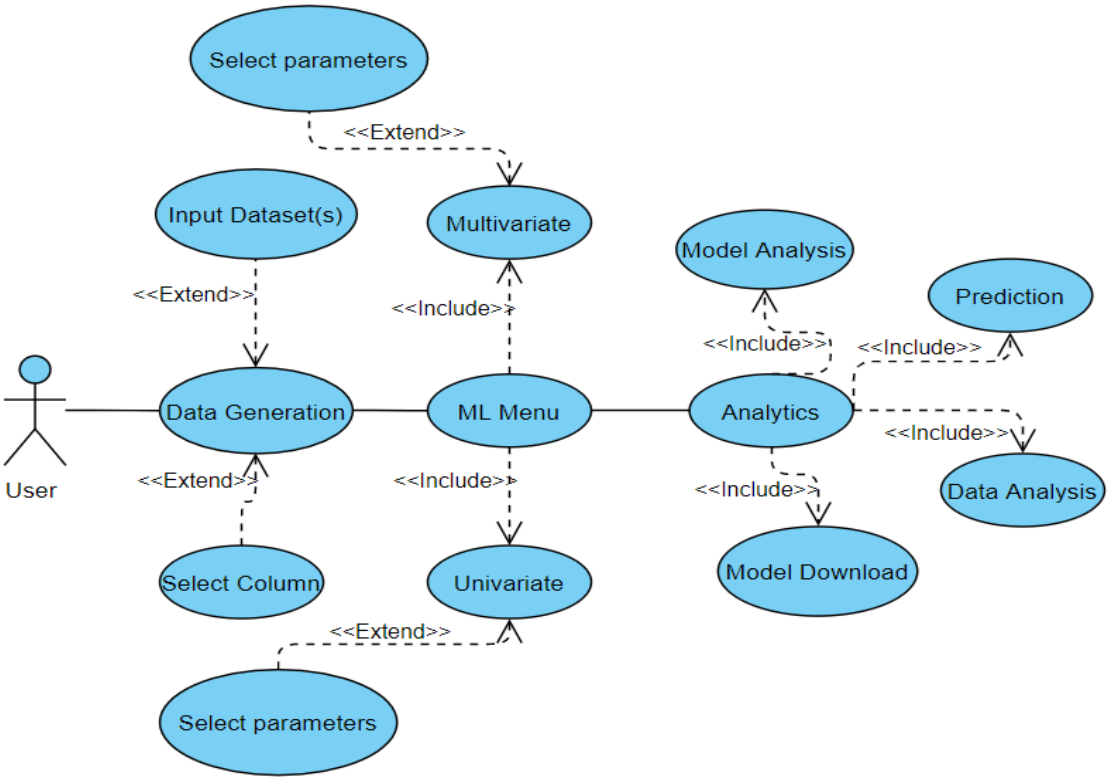


Figure 5 - Dataset creation with ML use case.

3.2 Risk Assessment

With a full set of requirements, a risk assessment can be had. This assists in identifying risks followed by mitigation to avoid unnecessary delays/problems [87].

Focusing on software development, Table 7 displays the top risks with solutions if one of these occurs for the remainder of the research. Table 7 includes the Probability (P) of a risk occurring, the Impact (I) the risk will have and lastly the Exposure (E) of the risk, calculated by P x I.

#	Risk	Description	P	I	E	Reduce risk
1	Schedule Creation	- Delay in one task occurs ripple effects on dependent tasks.	8	3	24	Pre-task, ensure requirements/methods are approved by those involved. Ensure appropriate time scales are implemented for each task with some leeway in the event of a delay.

		- Re-estimation to tasks that occurred delay is overly confident.				
2	Security Of Data	- User data is compromised.	3	9	27	Ensure password encryption is of the utmost standard.
3	Scalability	- Research is not valid for large scale production.	4	10	40	Provide adequate simultaneous testing and use technology that adheres to large scale productions.
4	Redundancy	- Technology/ methods are considered redundant.	2	6	12	Use technology that is recent and still backed by high corporations.
5	Complexity	- Applications are too complex for the targeted user.	7	8	56	Human-Computer Interaction methodology will reduce this impact.
6	Application/ files corrupted or lost	- Event of data necessary for development being lost.	2	10	20	Use of cloud storage to mitigate the impact of this occurring along with frequent backups.
7	Maintaining	- Not able to maintain postproduction.	6	5	30	Using computer standards such as documentation and comments to assist future developers.

Table 7 - Risk Assessment

The three systems are to be designed and implemented in a clear manner to accomplish the objectives that were mentioned in section 1.2. These three models will work uniformly to allow simple, fast, and efficient production, as displayed in a simplistic view in Figure 6. These modules are: (1) the first module deals with the collection of data and reporting on data using a mobile device for mobility, therefore, this module is named Collective Reporting on Mobile (CROM), (2) the second module deals with a more advanced overlook analysis from the data collected through CROM, therefore, this module is called Reporting Analysis Web (RAW) application. (3) the third and last module deals with dynamic dataset generation from remote multiple streams for on-the-go prediction using the AI RNN engine (DOSAN).

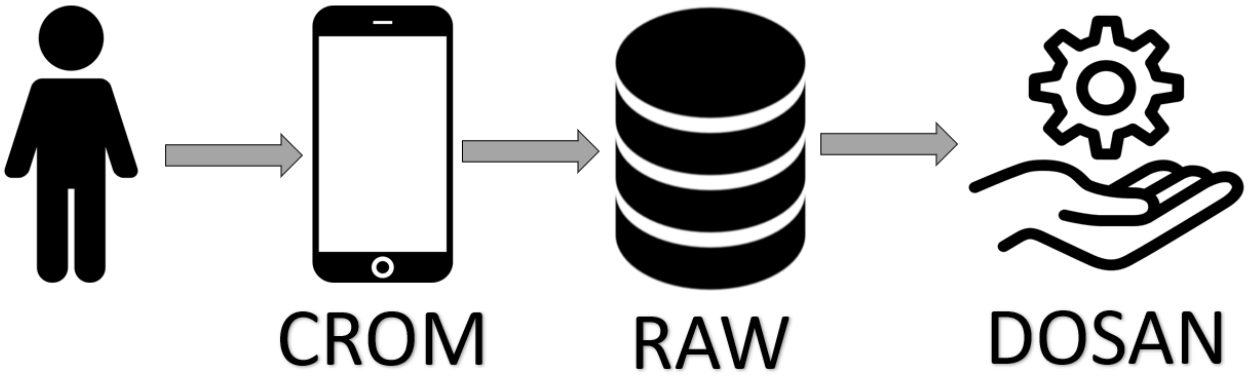
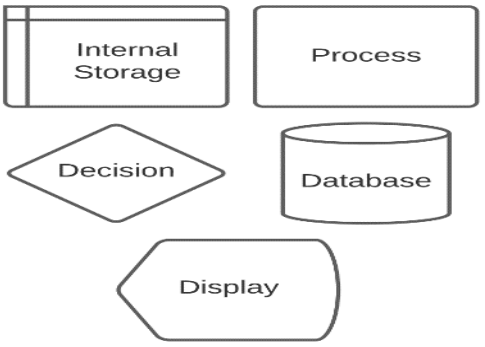


Figure 6 - Simplistic view of uniformed process.

This chapter includes the following for each module:

- Product Breakdown Structure (PBS), which is a tool to illustrate the components needed to compile the overall product/software. This assists in a clear understanding of what is required.
- The product Flow Diagram visually represents the necessary process for establishing the final product. Using graphical representations assist in the development of the product as there is an easy-to-follow path. A key is provided in Figure 7 to show the meaning of each shape.



- Skeleton Design of the UI to mimic the layout of the page without showing actual content. This wireframe assists in the fast deployment of the UI as it means all stakeholders will give consent before implementation.
- Database Schema, the same schema is used for CROM and RAW, representing the logical configuration of the database, accomplished in a visual representation to demonstrate constraints and requirements.
- Finished module, demonstrating the result of the module and any key attributes to mention.

3.3 CROM

Module 1 primarily focuses on the user inputting data along with transparent reporting. The first step before building CROM is looking at what components are needed; therefore, a PBS is created to assist in the development based on the Use Case diagram from chapter 2.2.2.

3.3.1 Product Breakdown Structure of CROM

CROM is broken up into the following components as displayed in Figure 8:

- Front End – This component encapsulates elements that make up the front end. There are four subcomponents which are Authentication, Input, Reports and Weather Analysis; Authentication consists of Login and Sign Up, Input includes QR scanning and Manual Entry, Reports is a collection of Manual Entry, Trap Entry and Trap Maps and lastly is Weather Analysis
- Back End – The elements that make up Back End are Database, Google Services which include Weather and Map and lastly is Mobile Services which has Camera, GPS, and Network.

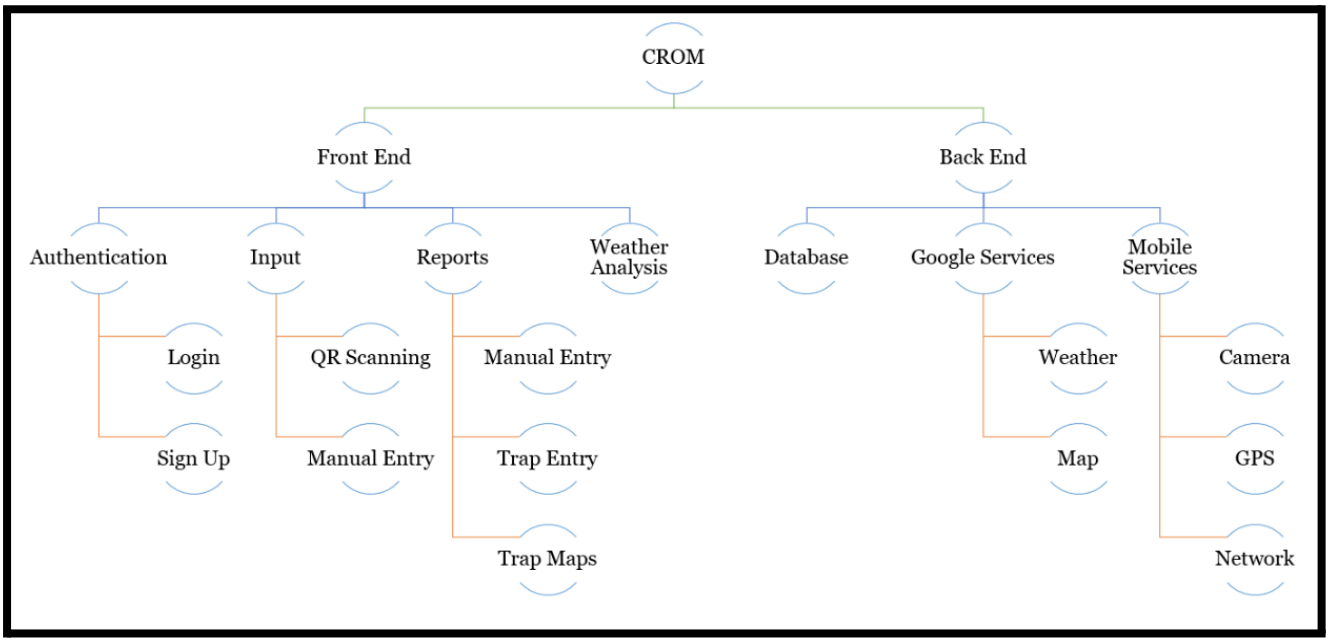


Figure 8 - Product Breakdown Structure of CROM

3.3.2 Product Flow Diagram of CROM

After concluding the PBS, the following step is the Product Flow diagram in Figure 9. The flow of CROM should be that the user can “Sign In” and go to authentication, meaning if there are no details, then giving the option to “Sign Up”. After successful authentication, then a menu displaying a series of options are available for selection. Two options, “QR Scan” and “Input”, will input data into the cloud database whereas the other two options, “Trap Map” and “Report” will be displaying information based on data from the cloud

database. The final option is “Weather”, displaying the current weather report by linking to mobile phone services such as GPS.

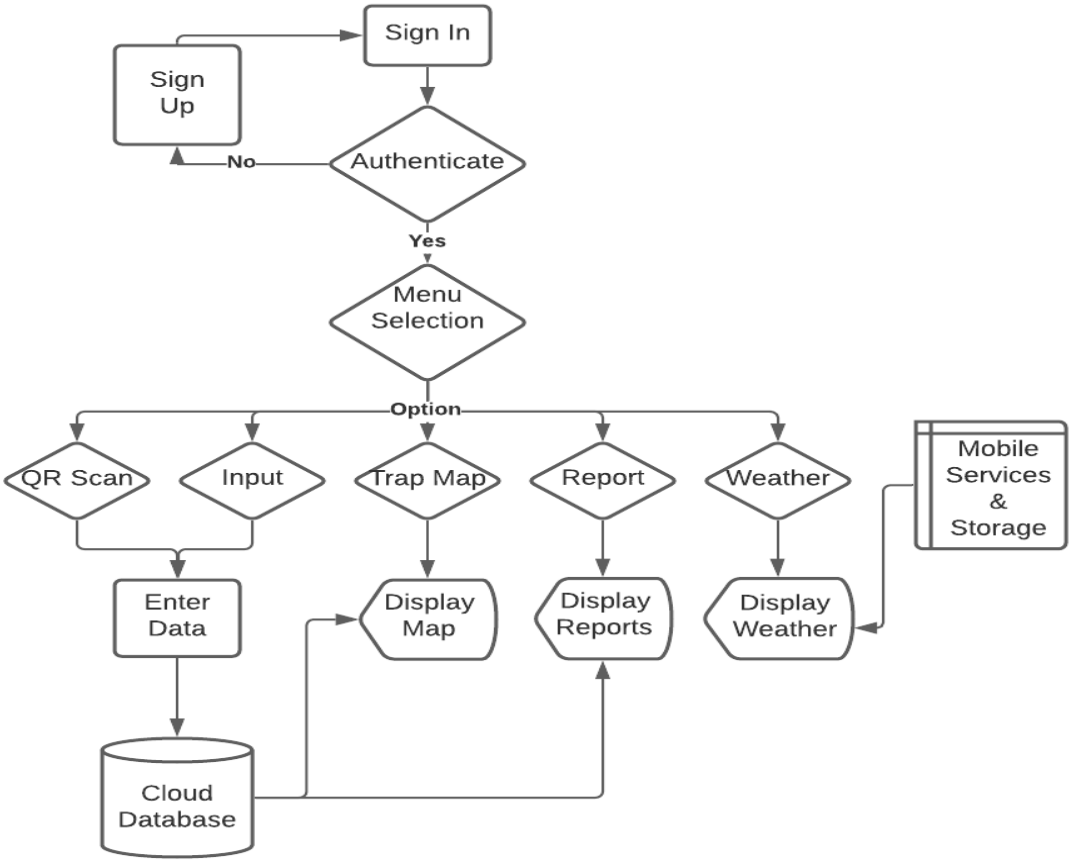


Figure 9 - Product Flow Diagram for CROM

3.3.3 Skeleton Design

The skeleton design process compliments the activity flow by showing how the screens are in order from left to right. The first three screens, as displayed in Figure 10, will be for logging in, followed by signing up - if no details are on record, and then bringing the user to the menu. Following from the menu, the next pages available are the input which leads to reporting and map viewing.

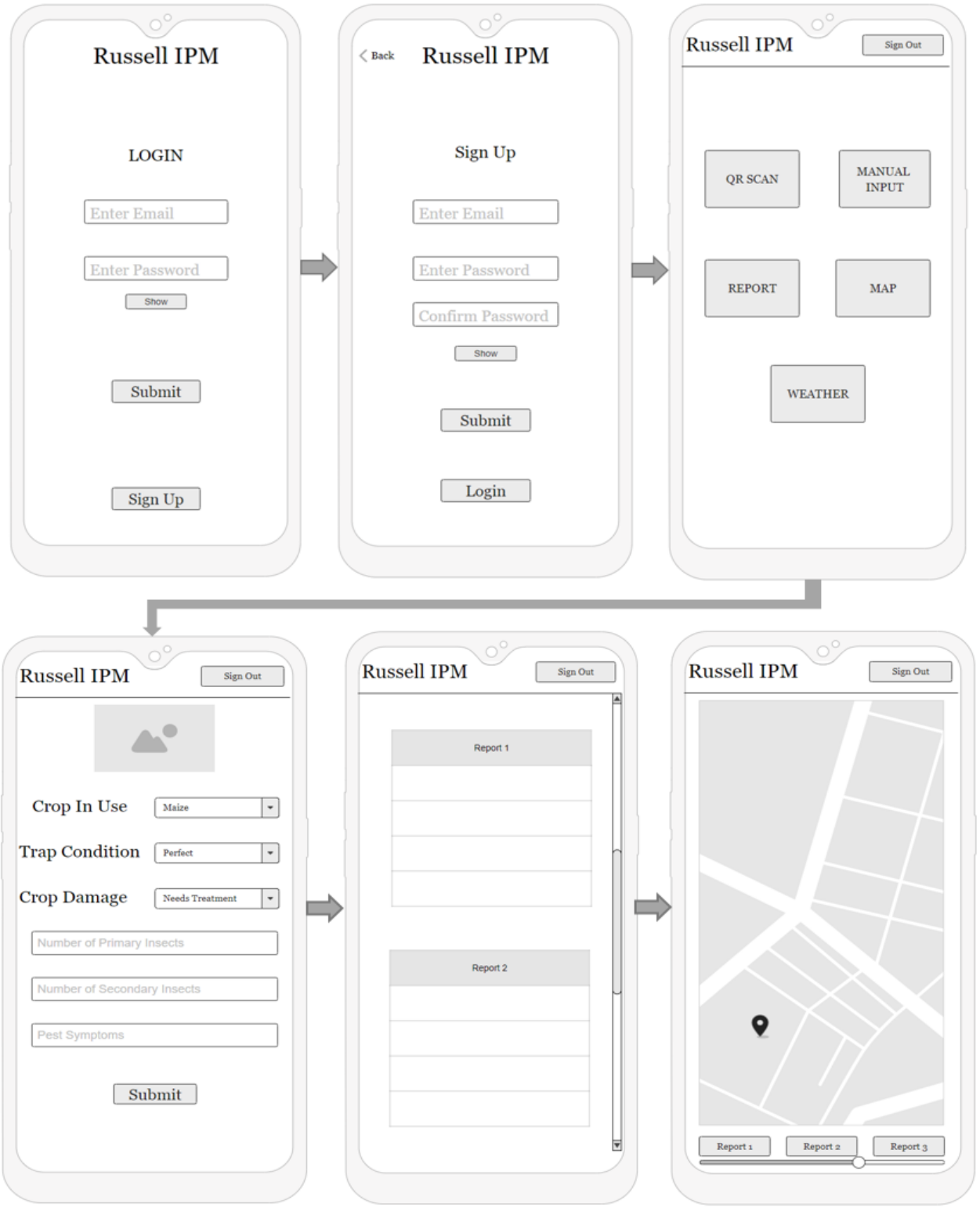


Figure 10 - Skeleton of CROM.

3.3.4 Database Schema

There are three databases, User, Image Storage Bucket and Reports, displayed in Figure 11. The User database stores all user information, being Name, Email and Password as strings

and lastly UserID which is an auto-increment field. The UserID is linked to the Reports database which stores information that is generated through CROM. The fields in the Reports database are:

- UserID – The primary key.
- Crop – String value focused on the type of crop.
- Crop Damage – String value for identifying the level of damage had on the crop, for example, if the crop is extremely damaged then it would be classified as, “unusable”.
- Pest Symptoms – Integer based on the fact users are to submit numerical values of how many leaf mines there are in the crop which tells an entomologist the symptoms being shown.
- Primary Insect – Integer as this is a numerical count of the majority insects on one crop.
- Secondary Insect - Integer, like Primary Insect however this is an optional requirement and is a numerical count for the second majority of insects.
- ImageReference – Foreign key, this is a String value as it will be automatically generated when the user inputs a photo and assigned a String based on size, date, and method.

The last database is Image Storage Bucket which is using the ImageRef as a primary key that is being linked to Reports and be used to store the images associated through the field Image.

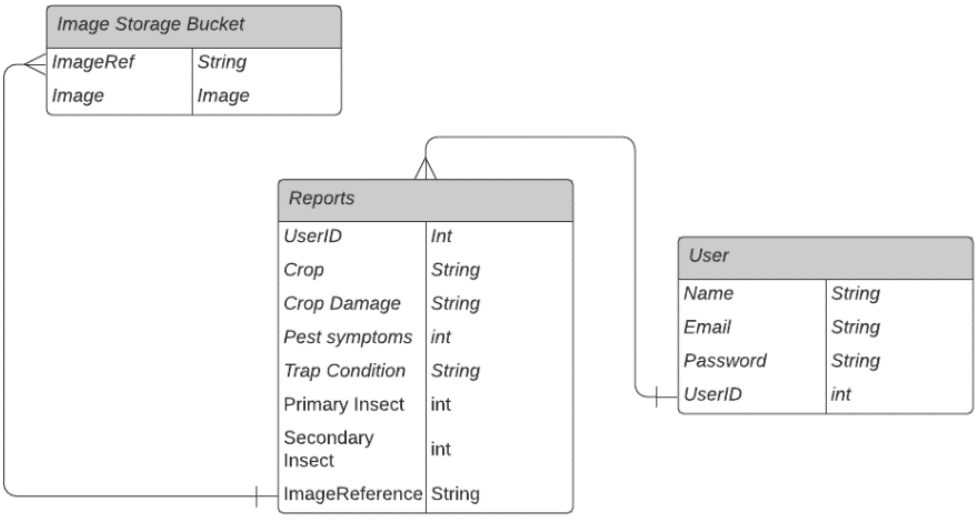


Figure 11 - Database Schema for CROM & RAW.

3.3.5 Finished Module

The final implementation was completed within a month and a half. The final version of the module, as displayed in Figure 12, concluded using a colour palette of light blues with greens, to compliment the colours of the sponsoring company. CROM required only Firebase, Flutter, and the use of API's, discussed in chapter 2.2.1.

The start of the application opens to a welcome screen with two buttons, "Login" and "Sign Up" for easy navigation. The "Login" page is identical to the skeleton design. The "Login" page will demonstrate the appropriate error message if the user details are incorrect along with the option to sign up or a forgot password button. Clicking the forgot password button will send an email to the linked account with the option to change the password. The "Sign-Up" page also ensures errors are correctly managed, for example, if the passwords do not match or if the email is not in the correct format standards of an email, such as, "test@gmail.com". The only difference between the "Menu page" and the skeleton design is that the buttons include icons to represent the action along with another added button, "Analysis" which is to be implemented in the future to link to DOSAN results.

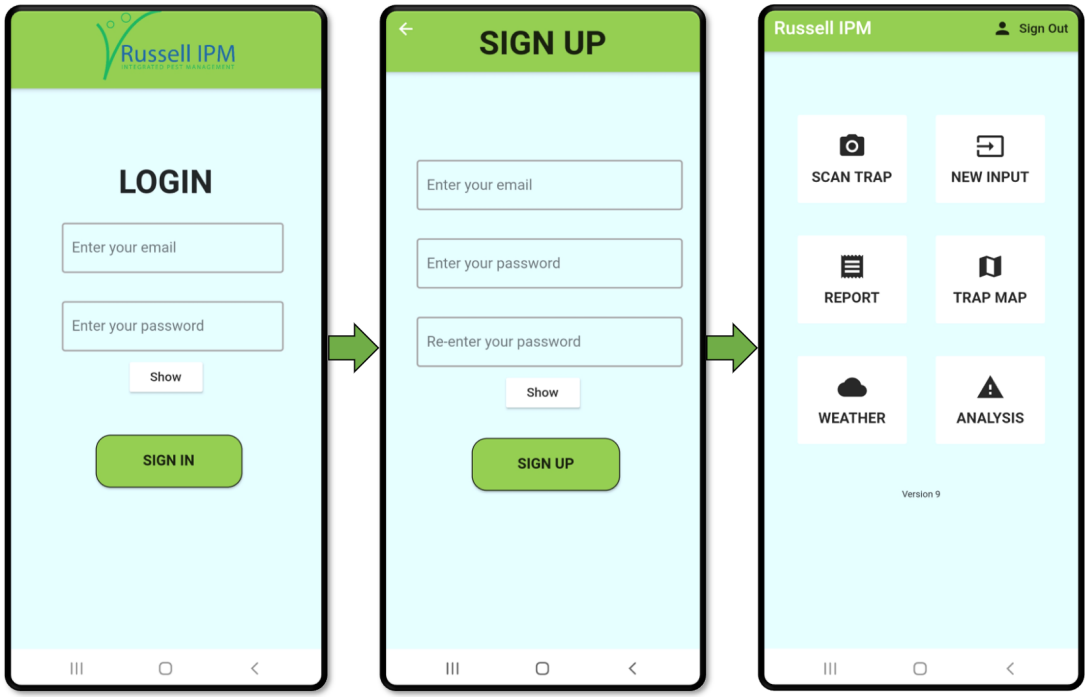


Figure 12 - Demonstration #1 for CROM.

Inputting for both "Scan Trap" and "New Input" are similar with the only difference is that "Scan Trap" displays the coordinates of the scanned item. "Scan Trap" checks that not just any QR code can be scanned, the page will only progress if the format of the QR code is in the format of latitude and longitude. The input fields are quite similar to the skeleton design with the only addition being adding an extra two camera inputs. The reason for the extra

images slots is so that if one image is blurry then the other two can be used still. These images would be incorporated with a CNN image recognition system to identify the insects in future expansions. The design still remains that, “Crop In Use”, “Trap Condition”, and “Crop Damage” are drop-down buttons for easy selection as displayed in Figure 13. The change for, “Primary Insects”, “Secondary Insects” and “Pest Symptoms” are that there is an icon added to the left of the title for some added design along with a,”?” icon button to display additional information into what the corresponding title means, an example of this can be found in Appendix 18. Since “Primary Insects”, “Secondary Insects” and “Pest Symptoms” are numerical inputs, the keyboard for these entries is limited to only show the numerical keyboard of the phone, as displayed in Appendix 19. Error handling has been implemented to ensure data cannot be submitted if (1) the three image slots are empty, or if (2) the “Primary Insect” and “Pest Symptoms” is null. Upon successful submission, the user will be displayed a data saved screen informing that the data has been submitted with the time and location of the submission (This is using the phone GPS to determine where the user was when data was submitted which is then able to find the postcode that can be used for analysis).

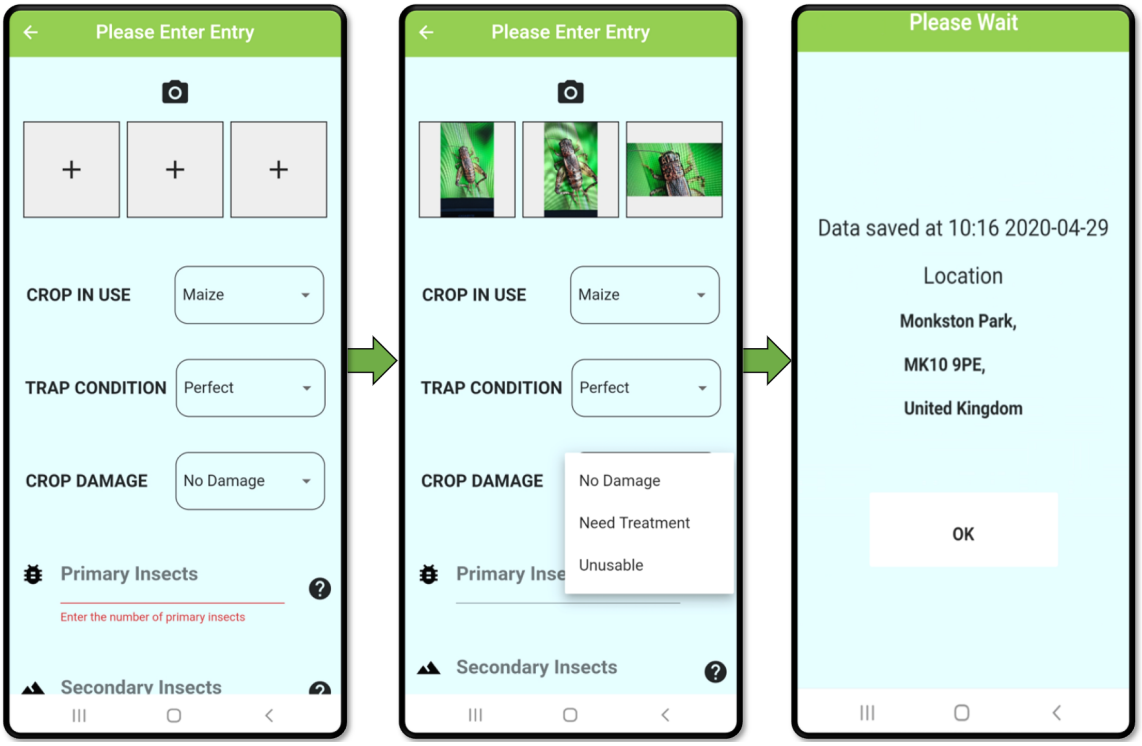


Figure 13 - Demonstration #2 for CROM.

Reporting can now be done as the data has been submitted. The reporting is low level, so it entails a summary of all the user’s submissions. However, this report includes the conditions from when the submission was inputted, such as location, temperature, and humidity, as displayed in Figure 14. The reporting page is dynamic with an infinite scrolling page that will automatically update whenever a new submission is acquired. If there is no

data for either Manual or Trap, then a loading widget is displayed with a message informing the user to ensure data is submitted. The trap map is an added feature to visualise where a trap has been scanned along with card details to give insight on the latest entry for a trap. This map, like the reporting page, is dynamic and will add points to the map automatically when data is submitted. An additional feature is that clicking the trap in the card, it will then move the camera to that point of interest. The weather page was added as an additional feature since the functionality was already in place for the backend. The weather page displays the user's current weather in their location. This information includes:

- Wind – The speed in Miles Per Hour (MPH).
- Pressure – Displayed in hectopascal units.
- Humidity – Shown as the percentage of humidity in the air.
- Sunrise – The time sunrise occurs.
- Sunset – The time sunset occurs.
- GeoCode – Displaying the geographical coordinates.
- Max Temperature – The maximum temperature in degrees °.
- Minimum Temperature – Display the lowest temperature in degrees °.
- Country – Showing the country code.

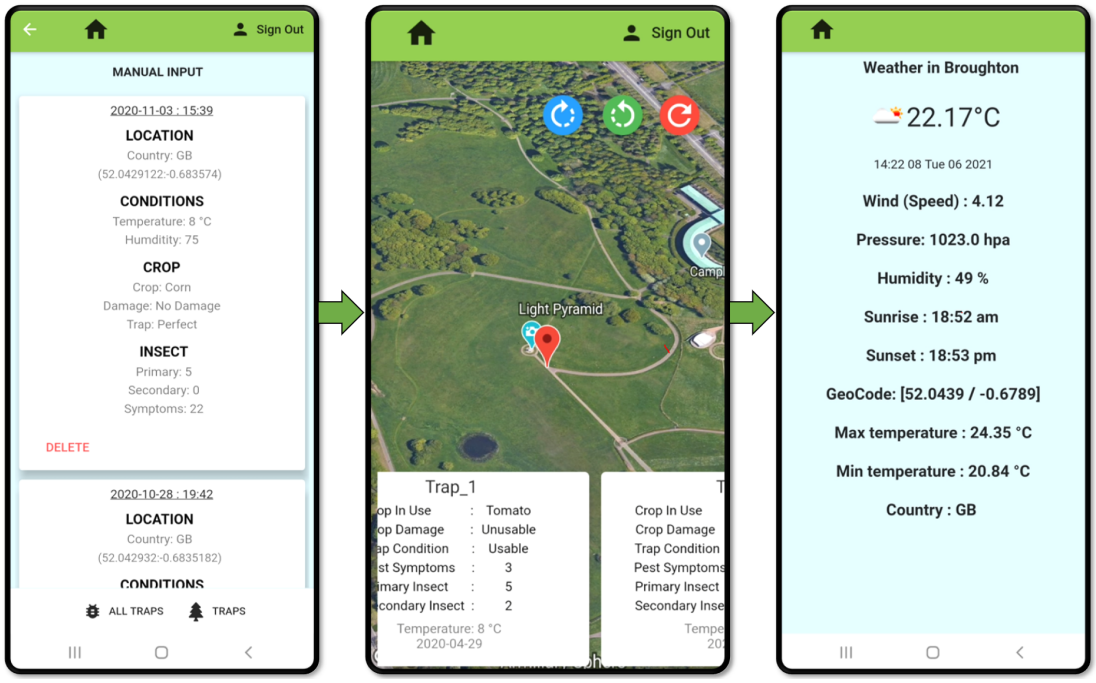


Figure 14 - Demonstration #3 for CROM.

3.4 RAW

Module 2, RAW, focuses on reporting the data from CROM at a higher level. The PBS is created with the assistance of the requirements and the Use Case diagram from chapter 2.2.2.

3.4.1 Product Breakdown Structure of RAW

RAW can be broken down, as Figure 15 displays, into the following:

- Front End: incorporating Authentication with Login, Reports with Manual Entry and Trap Entry and lastly, Statistics with Manual & Trap Bar chart, Search Data Via Date, Symptoms Per Trap and Select Variables.
- Back End: Only Database is needed for the build of the Back End.

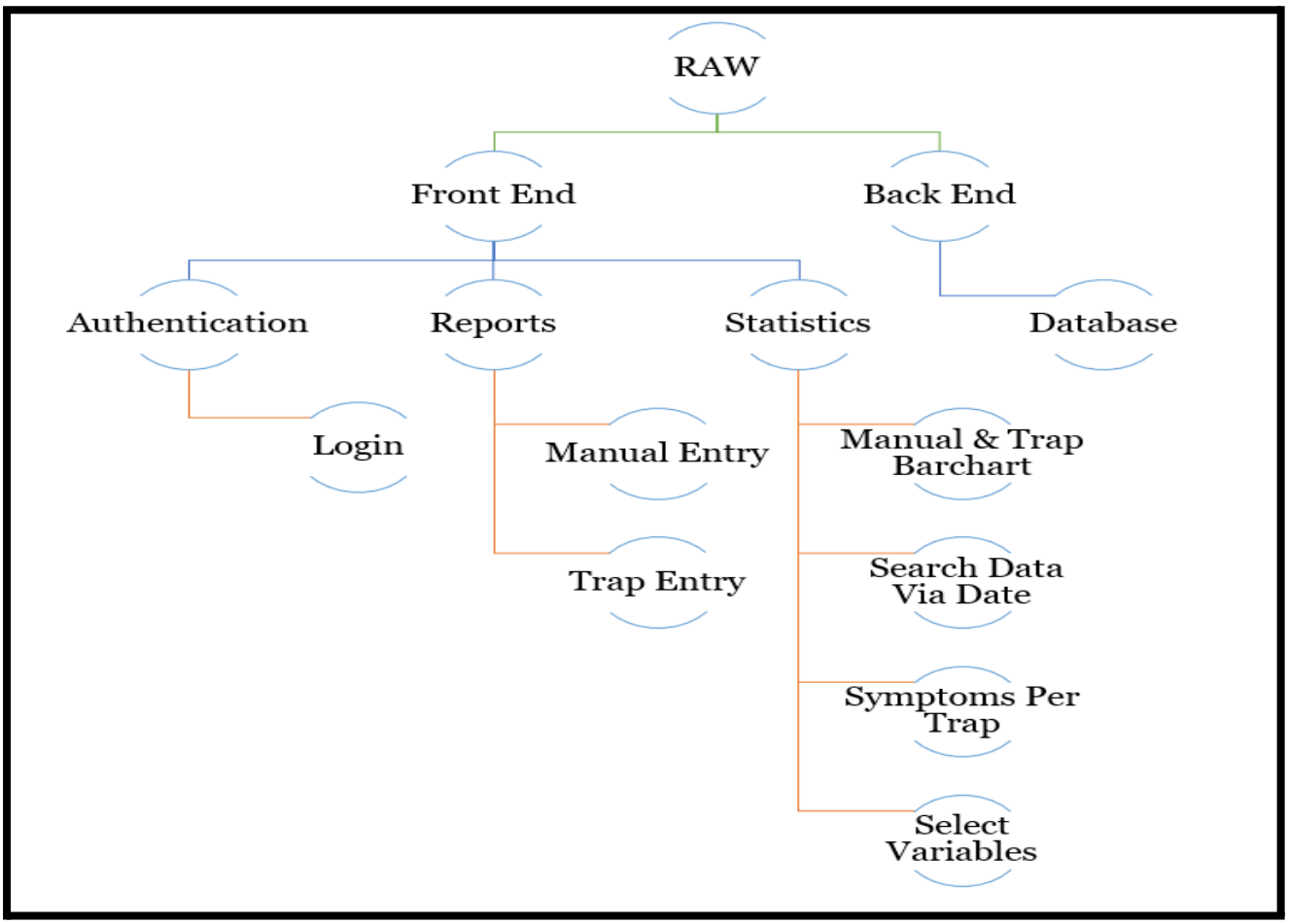


Figure 15 - Product Breakdown Structure for RAW.

3.4.2 Product Flow Diagram of RAW

The process of RAW, as displayed in Figure 16, is that a user can “Sign In” which will follow to authenticate, this either shows a warning message informing to sign up or retry. Once authenticated, the menu will display with the option to select “Reports” or “Statistics”. These two fields will display information linked to the database.

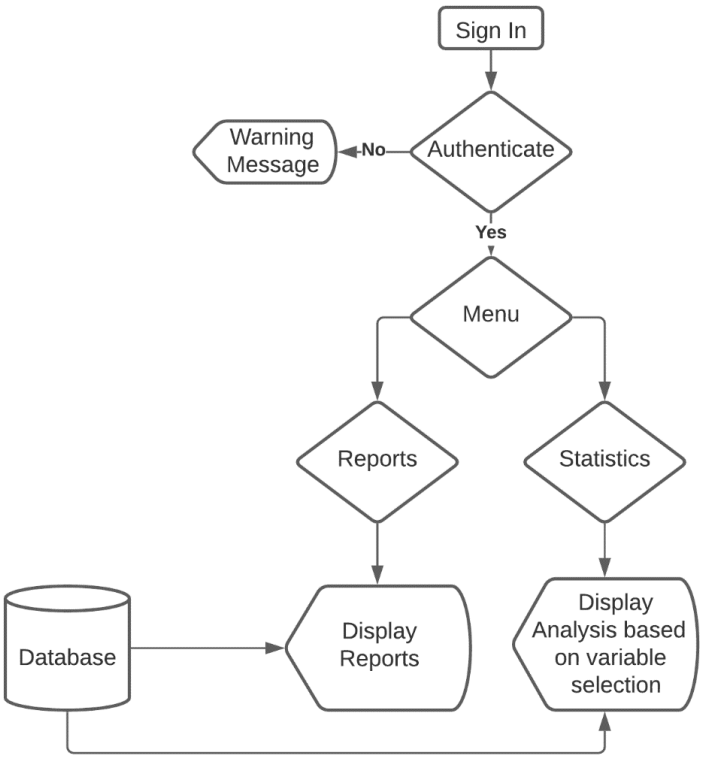


Figure 16 - Product Flow Diagram for RAW.

3.4.3 Skeleton Design

The first screens, as displayed in Figure 17, will be for authentication, following onto the menu for decision. The reports page is designed similar to the CROM report page, this being dynamic card generations linking information from the database with the option to cycle through “Manual Entries”, “Trap Entries” or “All Data”. Statistics, the other option from the menu, displays a further subset of options to choose from. These options include:

- Barchart.
- Trap.
- Search Trap.
- Search Manual.
- Symptoms Per Trap.

- Data VS Data.

Figure 17 - Skeleton interface #1 for RAW.

An additional feature page is the ability to search for data by dates either manually or a date picker entry, for example, either a specific date or a range such as the whole month of April as shown in Figure 18. The Data Vs Data page shows a line chart between the two variables to show any correlations along with analysis. The Barchart page too displays further analysis for reporting. Lastly is a pie chart displaying Symptoms Per Trap page, this pie chart demonstrates which trap has the worse symptoms along with the latest information on the trap.

Figure 18 - Skeleton interface #2 for RAW.

3.4.4 Finished Module

The final implementation of RAW was completed within a month. The final version of the module, as displayed in Figure 19 (Please refer to the appendix for larger images), had the same colour palette as CROM. As RAW is an extension of CROM, this meant that the resources were the same, this being Flutter and Firebase to build.

The Login page and Sign-Up page is identical to the skeleton design as well as the CROM Login and Sign-Up. Besides look wise, the error handling is also the same. The Menu page displays a welcome screen along with two buttons, “Reports” and “Statistics”.

Figure 19 - Interface#1 for RAW.

The statistics page offers a variety of buttons for further analysis as designed in the skeleton design. The bar charts are linked directly to the data from the database, resulting in a dynamic generation along with animation. The bar charts come with some added functionality which is that by clicking on a certain bar of the chart, then it will display

further information below the bar with corresponding data such as temperature and location for example. The search page allows for either manual entry of a date, such as 2020, or a date picker can be used. Appropriate error handling was had for the search page, meaning only date fields can be entered. The date from the search page is passed as a parameter into the analysis page to only generate the data with the same date, as displayed in Figure 20 where the parameter is set to 2020-04.

Figure 20 - Interface#2 for RAW.

Variable Vs Variable page offers a similar dynamic generation as to the bar charts, this being the animation and linking directly to the database. However, this visual representation is a line graph to demonstrate the smaller changes in a trend over a period. Dots are generated as date points to demonstrate where changes occur. Clicking on the dots will allow for further information according to that date, as displayed in Figure 21 where this example is between the trend of pest symptoms and humidity for 2020.

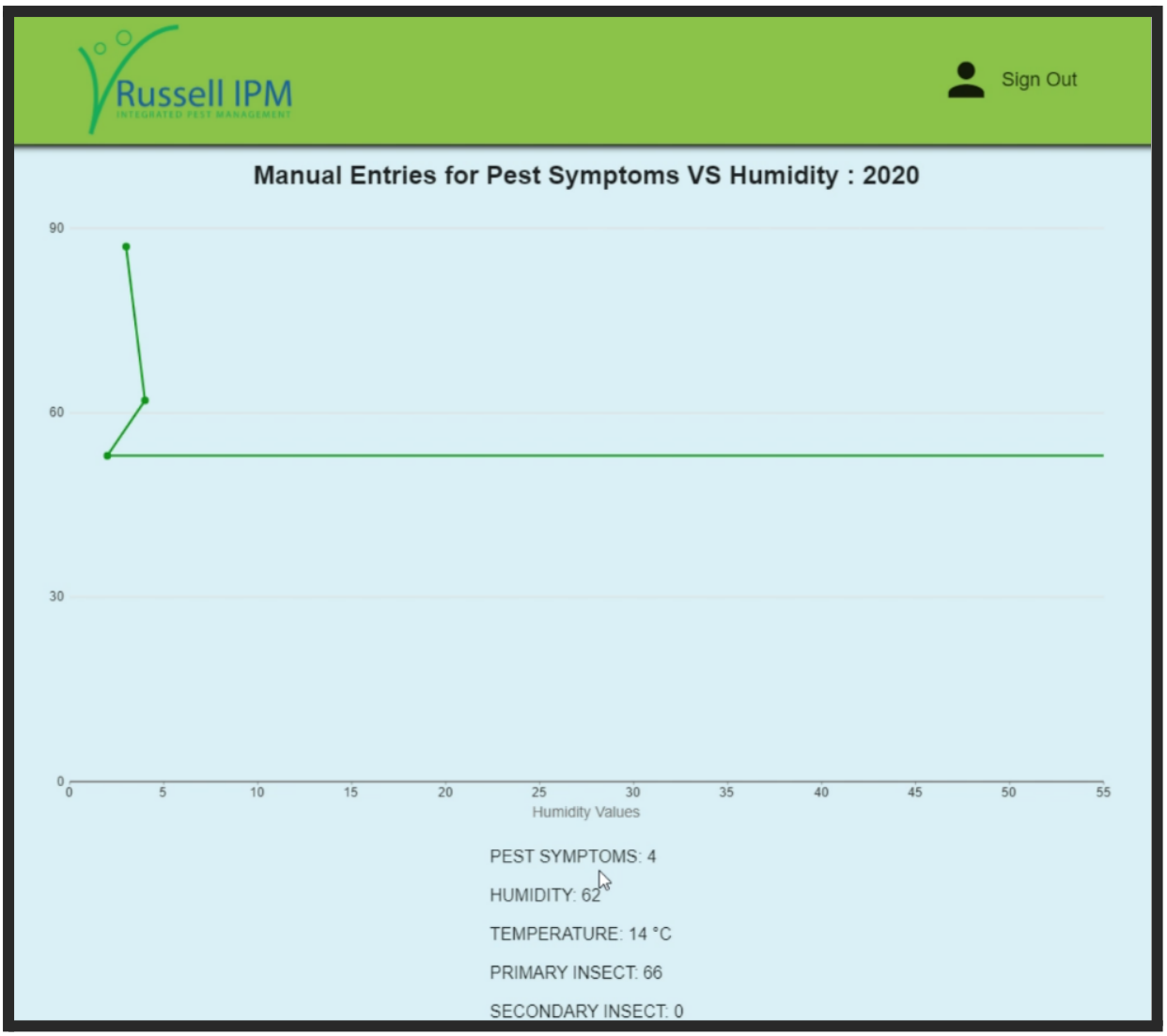


Figure 21 - Interface#3 for RAW.

Lastly is the pest symptoms page which visualises the traps through a pie chart to show which trap is worse in terms of symptoms, as displayed in Figure 22 (Please refer to the appendix for larger image). Along with just displaying, further information about the latest trap is added so that it can be used for future research and analysis.

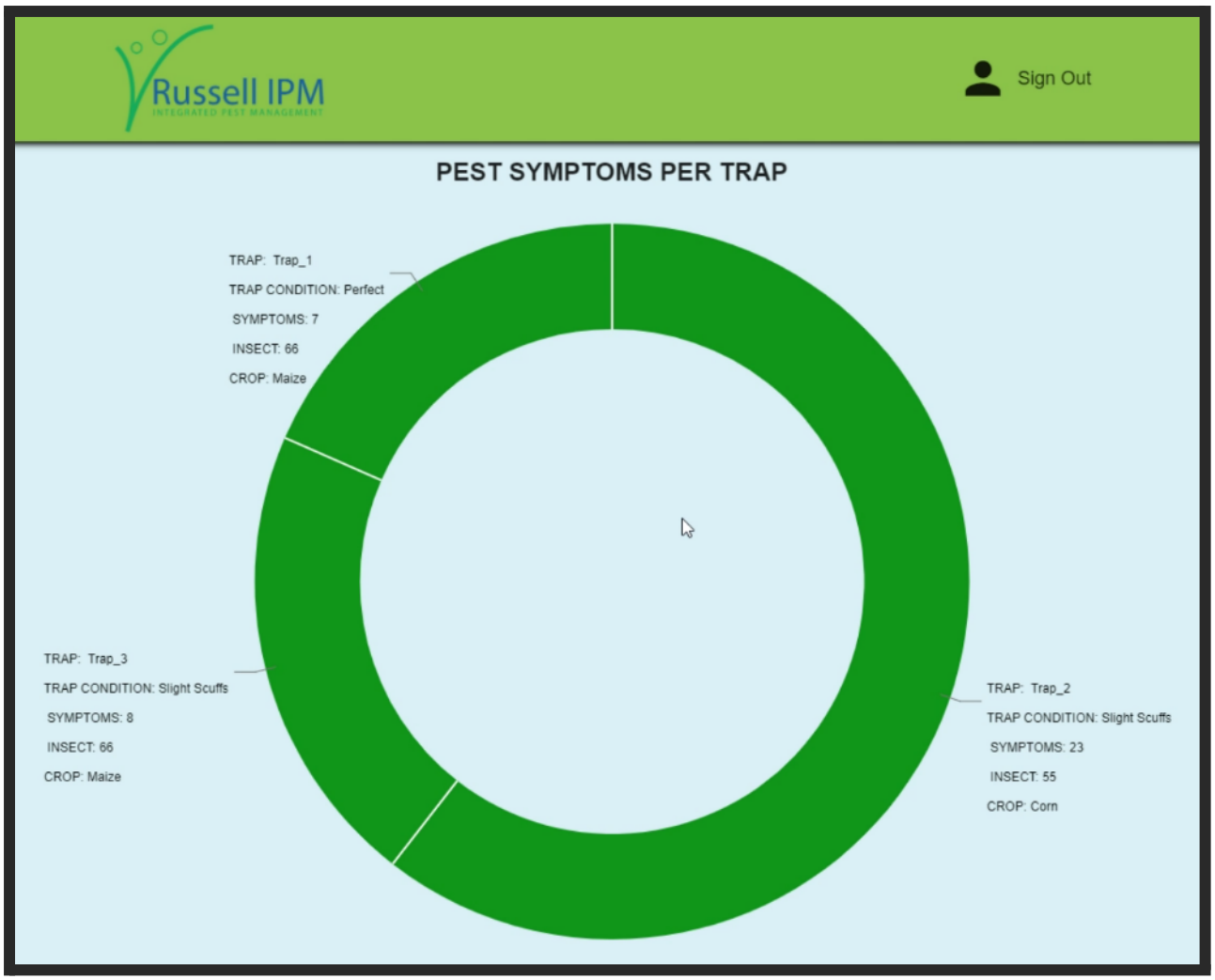


Figure 22 - Interface#4 for RAW.

3.5 DOSAN

Module 3, the primary focus of this paper, is to establish a system that allows for any number of datasets for an ML module prediction. Similar to CROM and RAW, the first step for implementation is creating a PBS which is based on the Use Case diagram from chapter 2.2.2.

3.5.1 Product Breakdown Structure of DOSAN

The PBS of DOSAN, displayed in Figure 23, is compiled of the following components:

- Dataset Generation: Making up from Column Selection and Input Dataset(s)
- Machine Learning: composed of a further subsection which is:

- Multivariate: Built by components of Model which includes Hyperparameter Selection, Test Split and Train Split. The second component is Data Select which includes Select Index and Select Variable(s).
- Univariate: In a similar manner as Multivariate with only one difference which is the Select Variable instead of Select Variable(s).

- Analytics: Model Download and Model Information make up Analytics. Model Information is made up from Loss, Root Mean Graph, Predictions and Baseline Predictions.
- Dataset Analysis: Compiled from Scatterplot and Trends. Trends is broken built from Y-Axis Selection and X-Axis Selection.

Please see appendix 28 for a larger image.

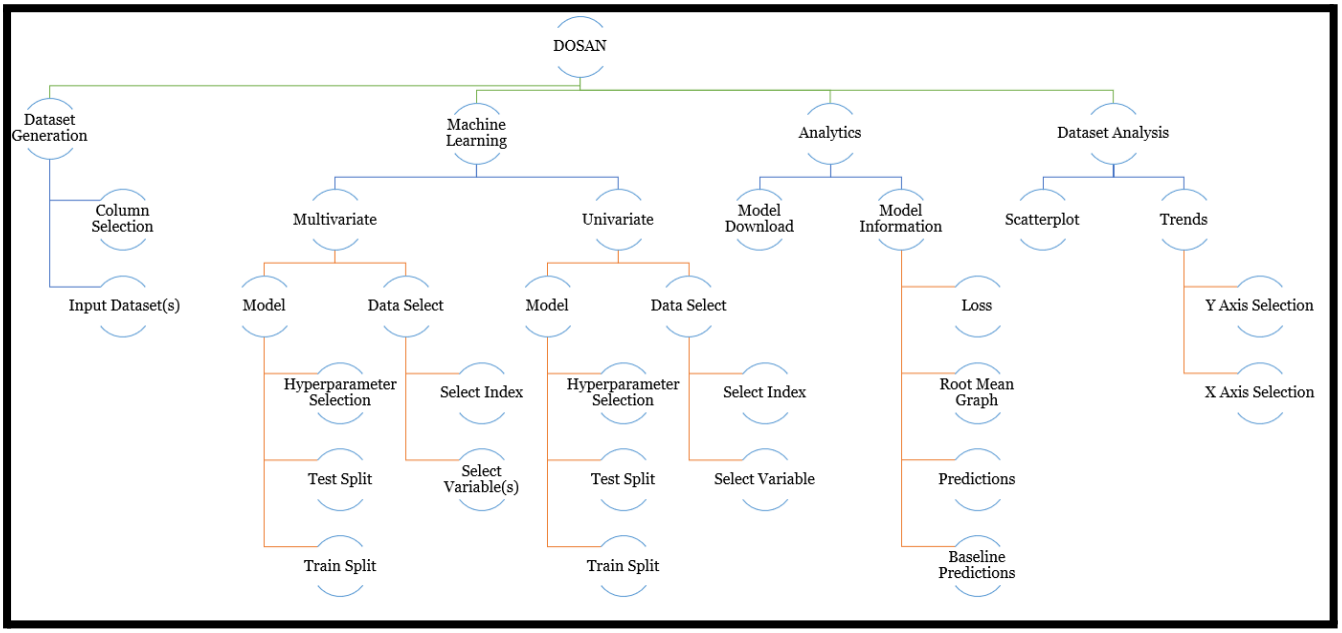


Figure 23 - DOSAN PBS.

3.5.2 Product Flow Diagram of DOSAN

Concluding with the PBS, the following step is the Product Flow diagram in Figure 24. The flow of DOSAN is that once the application starts the user will be presented with the "Input" screen to allow multiple datasets to be entered. Once datasets are entered then the process of combining the datasets is had. A menu is followed from the previous process asking what path to take, this being Univariate or Multivariate, these paths are quite similar but the only difference is that Univariate takes one index while Multivariate takes multiple. A decision to let the users select their own training and the testing split percentage is enabled as this can greatly affect the performance of a model due to either low data amount or even high amounts of data, as an example, if there are only 400 rows worth of data then a training split of 20:80 may result in poor accuracy (overfitting or underfitting) as only 80 rows are

testing 320 are for training, thus potentially resulting in overfitting. Not only can a user decide on factors such as train and split but also the hyperparameters or leave it at the industry standard defaults. The model begins training once hyperparameter selection is concluded, a console window is displayed to demonstrate the current training and used as a loading screen to show that progress is being had. Once training is concluded then the final page displays a list of options for analysis of the data and the trained model.

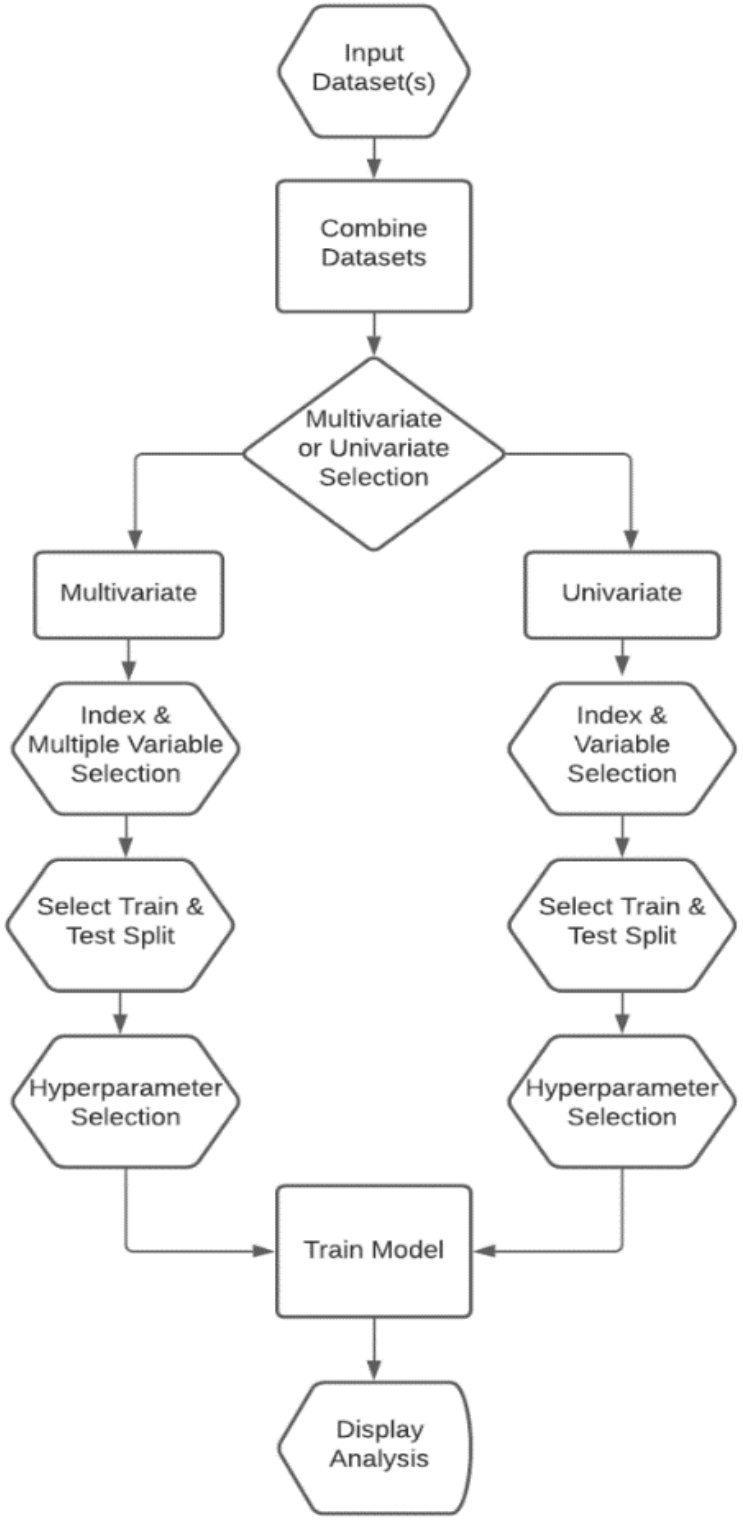


Figure 24 - DOSAN Product Flow Diagram

3.5.3 Skeleton Design

Starting the application will display a welcome title and two buttons, “Video Link” and “Start”. The link opens a YouTube video on how to use the software. The “Start” button will

open the next page, “Input Datasets” as displayed in Figure 25, where a button is utilised to choose files desired.

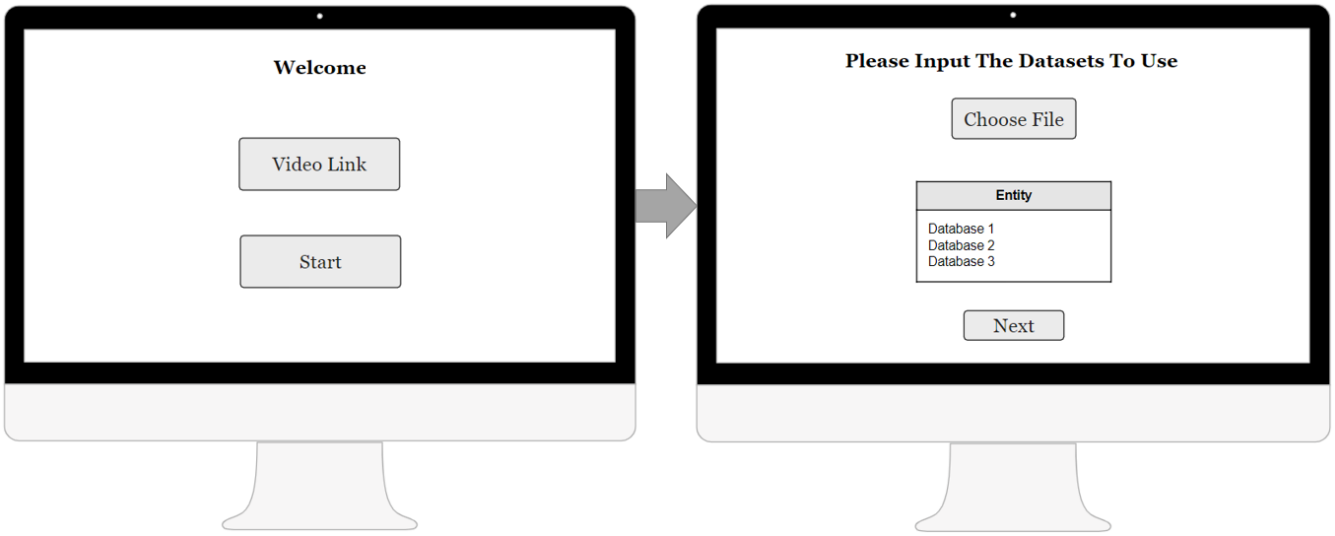


Figure 25 - Skeleton UI #1 for DOSAN.

Upon the user inputting the datasets to use, the next page/step is to choose what columns to be used for the final dataset and for training the model. The “Column Selection Page” is laid out with a dynamic button generation UI, this entails that the button will continue to populate the screen until all the column names are met, so for example if there were only 5 columns then only 5 buttons will be displayed. Submission of columns will lead to deciding whether to use Multivariate or Univariate which follows on to the “Choose Split page”, as displayed in Figure 26, where choose percentage buttons are displayed to allow easy train and split selections, as an example, there would be 50%, 60%, 70% and 80% for training followed by the same layout page but with 50%, 40%, 30% and 20% for testing.

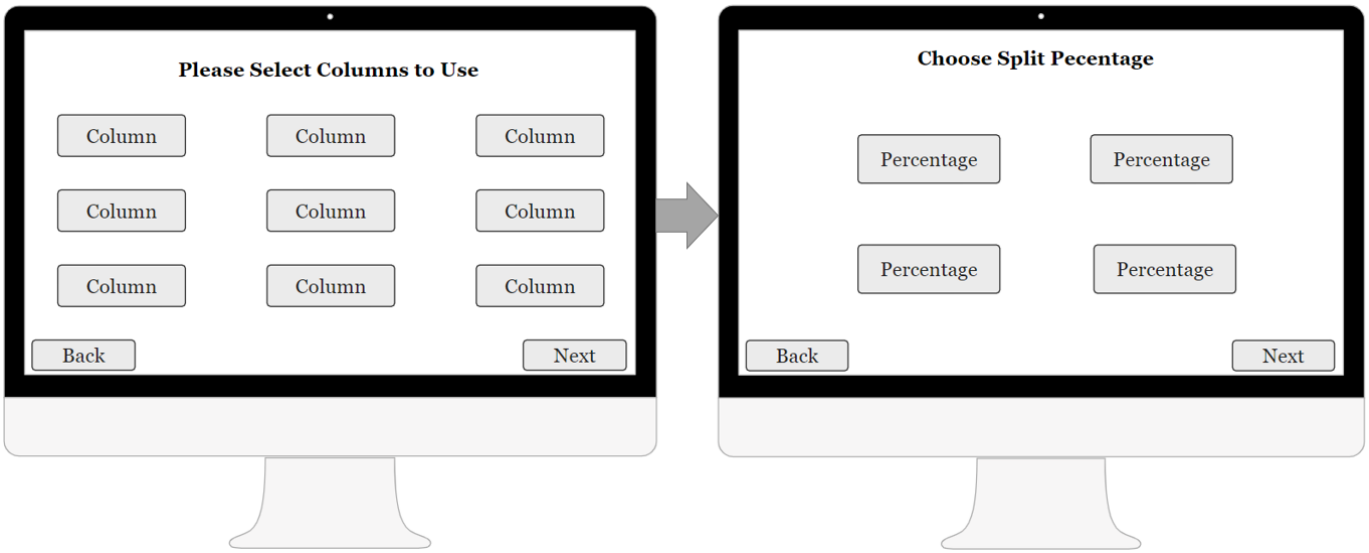


Figure 26 - Skeleton UI #2 for DOSAN.

Nearing the end of model pre-processing, the final activity is the optional tweaking of hyperparameters for the model. These customizations can be decided on values by using the dropdown button with pre-defined values based on ML standards as researched in chapter 2.1 of the background research. Finally, after the model is successfully trained, a "Predictions & Analysis" page is displayed to allow for a variety of options, as displayed in Figure 27.

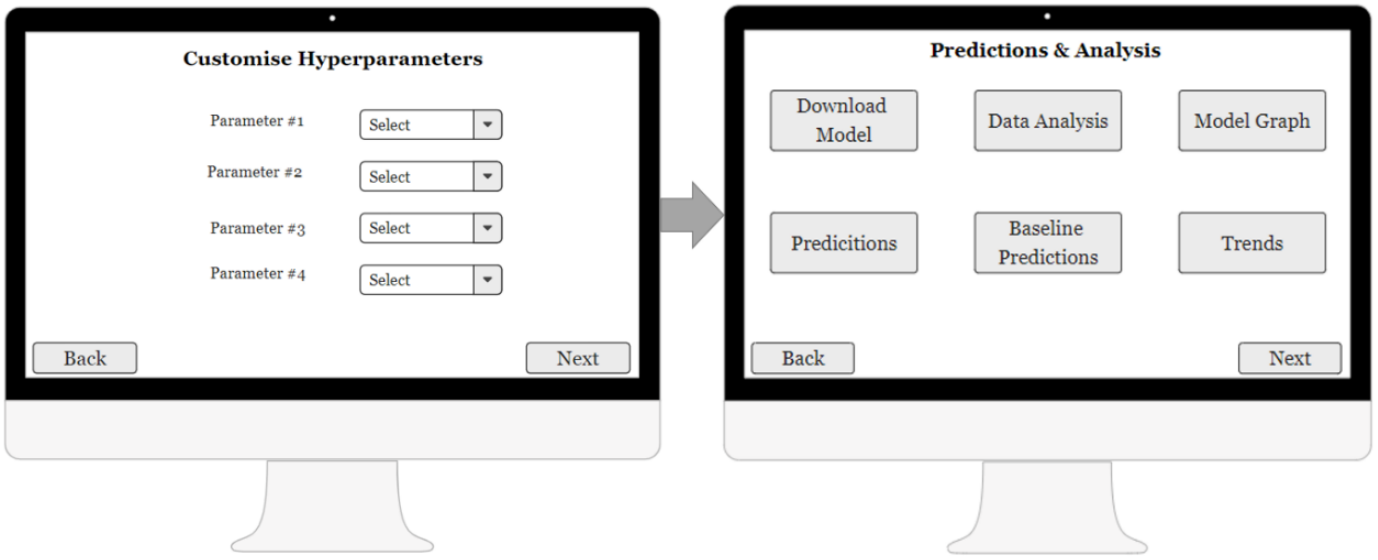


Figure 27 - Skeleton UI #3 for DOSAN.

3.5.4 Database Schema

Datasets of all manner can be used for compilation. DOSAN breaks down the datasets to only discover the column names that are then stored in an array using Pandas. Columns with the same names from different datasets are still stored as one single variable in the array and will be the foreign key used to find the connection between the datasets, this meaning that duplicate indexes are sorted from which one came first and removing the duplicate. For compiling the datasets, a minimum of one continuous foreign key must be had, for example, a date is a continuous data format. An example of this process, as displayed in Figure 28, is a WEATHER dataset that has data, "Rain" and "Temperature", relating to a location, Buckingham, and with one of the columns being date, 2020:06:01. The second dataset, CROP, has information, "Crop" and "Number of Crops" about a crop with a date of 2020:06:01. Lastly is a third dataset, INSECT, that has information, "Insect Name" and "Number of Insects" relating to a specific insect, this dataset has a date column of 2020:06:01. Therefore, the software can identify the similar data from the three and join all the data into one single dataset which can be used to find a correlation between the different datasets.

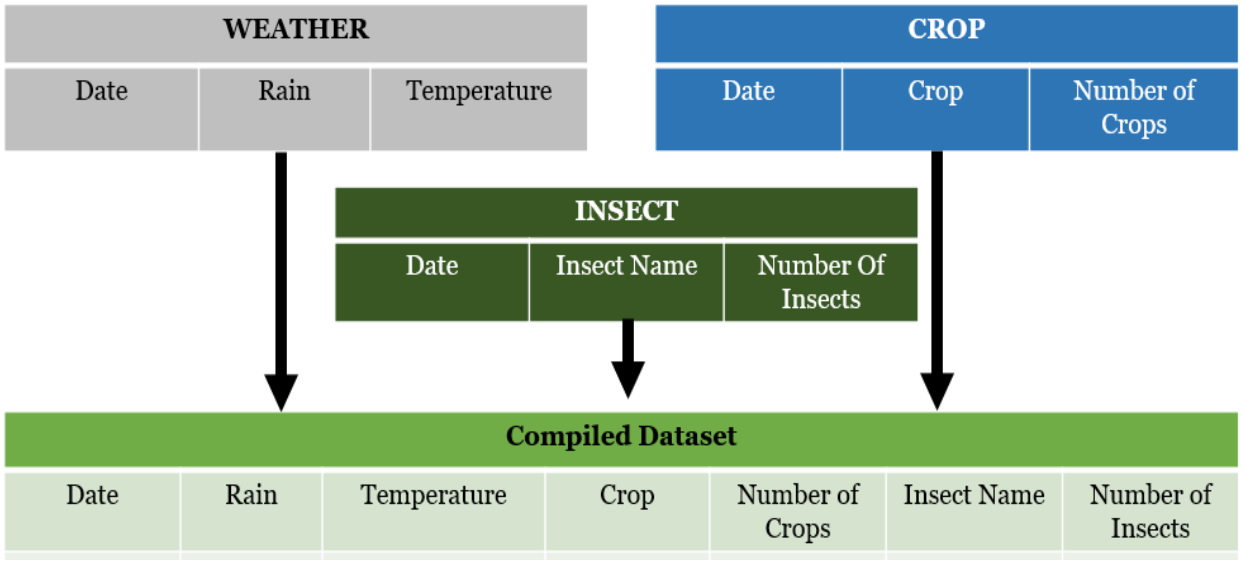


Figure 28 - Database Scheme for DOSAN.

3.5.5 Finished Module

DOSAN was completed within six months. The final version of the module had the same colour palette as CROM and RAW to maintain consistency. DOSAN required Matplotlib, TensorFlow, Pandas, Python, PyQt5 and Keras for the final finished build. The model was chosen and tested before the implementation stage for easier transferring and faster building.

Upon starting DOSAN, a starting screen displaying two buttons is had, one linking to YouTube for a video tutorial and another to begin the process of creating an RNN model. In the dataset input screen, there is a button linking to file explorer that has parameters set that only dataset type files can be used, for example, Microsoft Excel Open XML Spreadsheet (XLSX). A table to inform on the location of the datasets is updated whenever a dataset is chosen or removed, as displayed in Figure 29. Upon clicking "NEXT", DOSAN process two factors, (1) the columns of the dataset into an array that will be used on the next page and (2) the location of the datasets that will be used for final compilation when the next page is completed. The next page after deciding the datasets is the dynamic button generation function, displaying the columns in all the datasets but only showing one column of the same name columns. For example, if Date is had in one and Date is had in another, then the DOSAN will still recognise these as similarities and check that the formats match. In the event that date formats are not similar, such as "2020-04-15" and "15-04-2020", then DOSAN will reformat the data to a uniform standard, datetime64, done through the use of Pandas. The buttons will change colour to dark orange to demonstrate what has been selected, clicking on the button will unhighlight to show removal. The objective is to maintain little human error which is why users can only progress to the next process when input is had. As an example, in Figure 30, the user can only progress when (1) at least one dataset is entered and (2) when a minimum of two columns have been chosen for the new dataset with one of them being continuous data such as date.

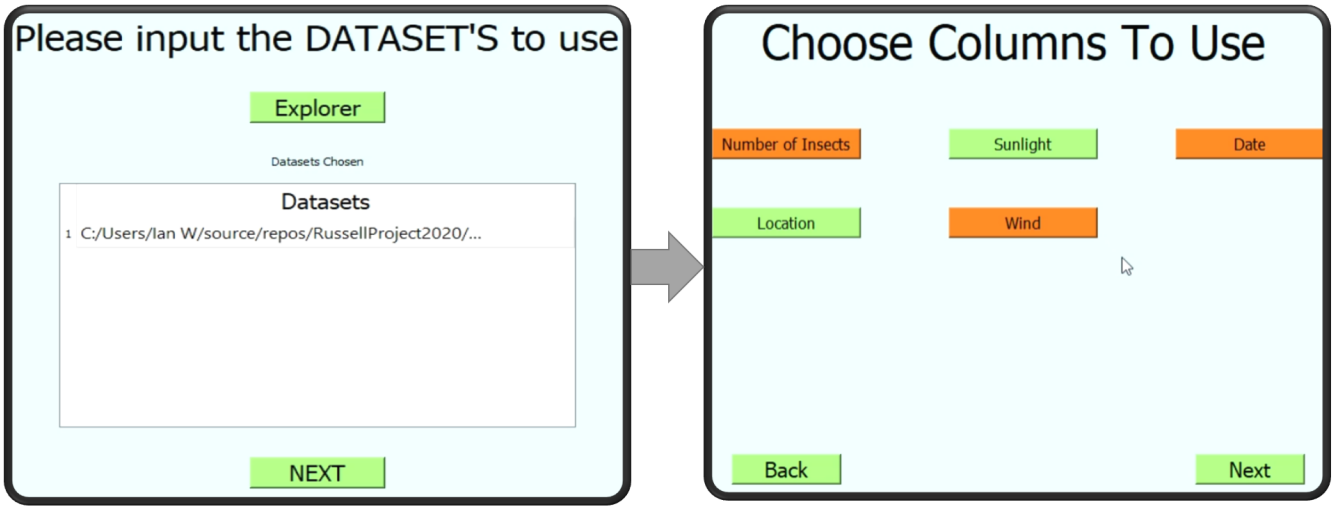


Figure 29 - DOSAN interface #3.

A new array, “A”, is generated with the columns from the newly compiled dataset. This array is passed onto the next page where users are given the option to select their “index” column, this requires the date to be in a continuous format therefore only columns with continuous data will be offered for selection. Another array, “B”, is created with the chosen column and is removed from the previous array. Therefore “A” = “A” – 1 while “B” = “B” + 1. Onto the next stage, the user again must decide what columns to use for focused targeted prediction, in this scenario in Figure X it is using Univariate so only one column can be used. This too means that “A” = “A” – 1 with “B” = “B” + 1, which will pass “B” throughout the stages to be used for ML training. Similar error handling is had regarding progressing to the next stage.

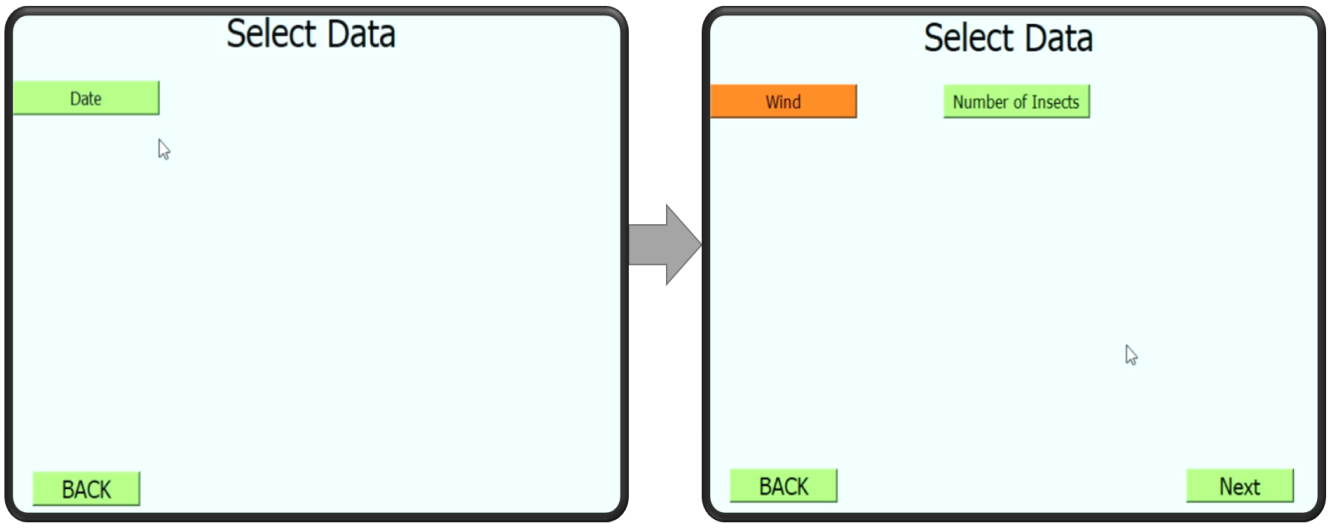


Figure 30 - DOSAN interface #2.

After the selection of index and target column has been chosen, then splitting the data is had. This is an added functionality to DOSAN, as studies from literature reviews in chapter 2.1 all utilise different splits. There is no standard to this and is subjectively based on the dataset. Allowing users to decide on the splitting for training and testing/history can greatly affect the accuracy for model prediction which will be covered in testing in section X. Following the design of the skeleton UI, the training is split into four buttons, 50%, 60%, 70% and 80%. When a training split has been selected then a check is had in the history split to decide which buttons will be visible for the user that makes up 80%. As an example, if 80% is chosen then only 20% can be chosen, seen in Figure 31, as anything above 25% will result in an error.

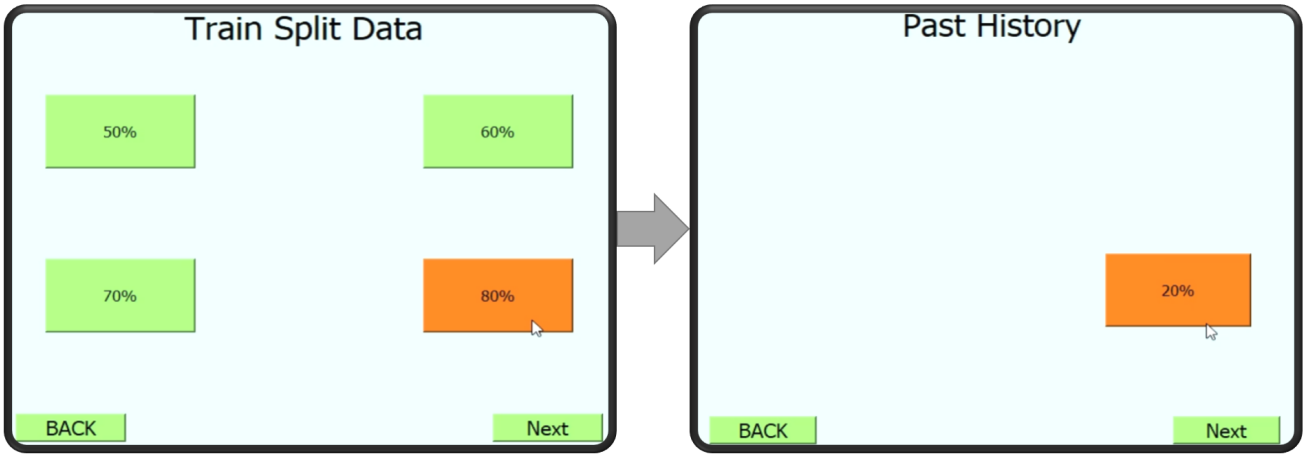


Figure 31 - DOSAN interface #3.

Nearing the end of pre-processing for the ML model, the final optional stage is customising the hyperparameters/parameters as shown in Figure 32. While some studies, such as those in chapter 2.1 of the literature review, state that changing hyperparameters is redundant and superfluous may be true, it is still worthwhile for testing such parameters that will be had in chapter 4.3. Therefore, due to the literature and complexity, this process is optional to the user. The values used in the drop-down are parameters that are typically used in ML development. Concluding with the button "Train", DOSAN will begin training the model with a time determined by the complexity. As soon as the model has successfully trained, the "Predictions" page is shown with the following list of options:

- Download Model – Ability to save and store the compiled model for future use in other applications.
- Data Analysis – Showing a histogram of numerical data that can be used to detect outliers or gaps from the dataset.
- Loss Graph & Root Mean Graph – These graphs visualise, using a line graph, the training within the model.
- Predictions – Showing an example of the performance and accuracy of the model based on the chosen target data along with the past data and the real future value.
- Baseline Predictions – Demonstrating the performance of the ML model by showing what prediction was had vs the real future value without ML.
- Scatterplot – Visualise the relationships between data had in the newly compiled dataset for further analysis.
- Trends – Additional functionality that allows users to decide on values for the x-axis and y-axis to visualise the current trend between the data.

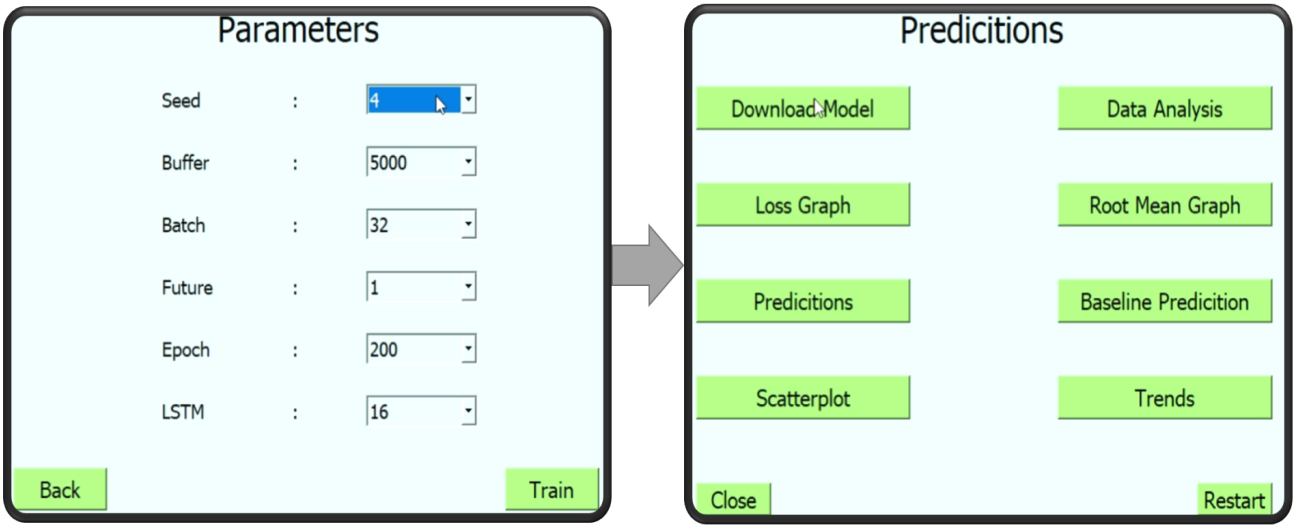


Figure 32 - DOSAN interface #4.

Chapter 4 Testing & Discussion

Testing is crucial for the software development life cycle to validate everything is working and achieving the objectives. Testing will be had for CROM, RAW & DOSAN. While multiple models can be used for testing, such as Agile [88] Waterfall [89] or V- Model [90] , the decision was to go with V-model to encapsulate the testing section for each system. The reason for choosing V-model is that this is an extension of the waterfall model where testing is made in parallel to the development [91]. A diagram of the V-model shows the process in Figure 33 [92].

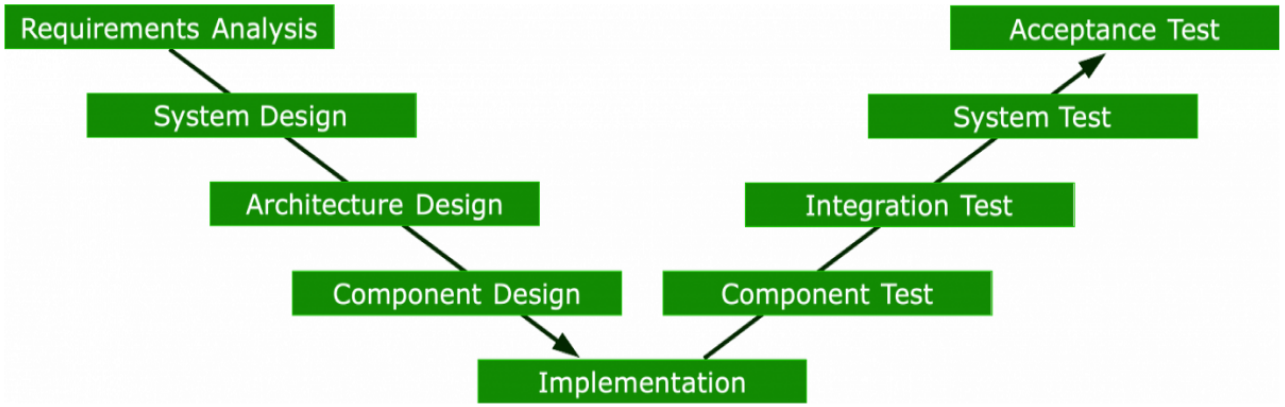


Figure 33 - V-Model Diagram

4.1 CROM

Focusing on the input, CROM testing looks at the application to evaluate completion. Testing was performed on Samsung devices (S6, S8, S10 & S20+, broken down in chapter 2.2.1 of the literature review).

4.1.1 Unit Testing

Unit testing is had to ensure every function/method is working correctly, as displayed in Table 8. This testing assisted in discovering and removing bugs in the software. To maintain consistency, five tests were had for each function to assess validity. The overall unit testing success rate was 79%, this is a very promising percentage as it highlighted the areas that required looking at more before progressing to the next testing stage.

#	Function	Description	Executions	Failures	Success Rate
1	Login	Ensure match is made from DB.	5	1	80%

2	Sign Up	Encrypt & check passwords match as well ensure the authentic email is used.	5	1	80%
3	Sign Out	Logs user out along with resetting cache so no possible re-entry without login/signing up.	5	0	100%
4	Scan QR Code	Camera opens in QR mode and checks the QR code is in the correct format before progressing.	5	2	60%
5	Button to Add 3 Images	Camera opens and saves images for Input (QR & Manual).	5	2	60%
6	Drop Down Buttons	Save selection for Input (QR & Manual).	5	0	100%
7	Primary Insects	Save selection & ensure correct format for Input (QR & Manual).	5	0	100%
8	Secondary Insects	Save selection & ensure correct format for Input o(QR & Manual).	5	0	100%
9	Pest Symptoms	Save selection & ensure correct format for Input (QR & Manual).	5	0	100%
10	Help Buttons	Ensure the popup widget is displayed when clicked for Inputs (QR & Manual).	5	3	40%
11	Submit Button	Check that required fields are entered for Inputs (QR & Manual); error displayed if not.	5	2	60%
12	Input to DB	Check that data from Inputs (QR & Manual) are saved in the correct format & correct times.	5	3	40%
13	Display Location on Submission - Inputs	Check users current time and location is displayed when submitting Inputs (QR & Manual).	5	0	100%
14	Manual Report Button	Check entries display in order by connecting from DB, if no entries then display a loading widget with a message.	5	2	60%
15	Trap Report Button	Check entries display in order by connecting from DB, if no entries then display a loading widget with a message.	5	1	80%
16	All Trap Reports	Check entries display in order by connecting from DB, if no entries then display a loading widget with a message.	5	1	80%

17	Trap Map	Check API links to Google maps along with linking to DB for adding POI for traps, along with camera starting on user's current position.	5	0	100%
18	Trap Map – Selecting Trap	Check if selecting trap animates the camera to the position of the trap.	5	2	60%
19	Weather Button	Get users current position and link to weather API for weather information.	5	0	100%
			95	20	79%

Table 8 - CROM Unit Testing.

4.1.2 System Testing

This is a set of finished, debugged tests before going to the next stage of testing/ completion. The use of system testing is to ensure the system completes the objectives defined [93] . Test cases focusing on the functionality of objectives are had in Table 9.

Test Case #1	User signs up followed by login then go to the "Scan Trap" button for QR scanning along with inputting data, once completed the user submits into the DB.
Objective(s)	Ensure users can (a) sign up successfully, (b) login by getting credentials from DB, (c) QR scan works with appropriate checks, (d) Inputted data (including data such as location and weather) is saved hastily in DB, and (e) Users current location is displayed upon successfully saved data.
Results(s)	The user was able to successfully sign up and log in with no errors, The QR scan opened the camera almost immediately and did the checks without any delay, meaning the page for input opened instantly. The only minor issue was that data at first was not being saved into DB but this was due to extending the Firebase access period (Currently only set to one year).
Test Case #2	Users will sign up, log in and go to the "Manual Input" button to open the page for inputting data manually that will submit into the DB.
Objective(s)	Ensure users can (a) sign up successfully, (b) login by getting credentials from DB, (c) Inputted data (including data such as location and weather) is saved quickly into the DB, and (d) Users current location is displayed upon successfully saved data.

Result(s)	Results were the same as Test Case #1, once the issue with the Firebase period was extended then there was no longer any issues. This fix was quick to do as the Firebase interface make it very user friendly.
Test Case #3	User logins, then from the menu to view manual & trap reports including the trap map.
Objective(s)	Check that (a) Login is successful, (b) Manual report page either displays an organised card format by order of the date that links to the DB for manual input or will display a loading widget depending if there is no data with an error message, then, (c) Trap report which is the same as (b) but linking to QR input and lastly, (d) The trap map will display a Google map of the user's current position with traps as a POI to select from that will move the camera to a position of trap.
Result(s)	There were no bugs found for Test Case #3.
Test Case #4	The user will log in to view current weather then sign out.
Objective(s)	Check that (a) users can log in with ease, (b) they can go to the weather button which will display the current weather of the user, finally (d) once the user is satisfied, they will log out.
Result(s)	The only hiccup in the application was that it took a second to connect to the weather API and get the users current location from their GPS. The fix for this was to pre-generate a location where the application is targeted for as the base weather which is then quickly overridden with the user's current location and weather details.

Table 9 - Test Case for CROM.

4.1.3 Performance Testing

The final testing for CROM is to examine the general performance of the system with the goal is to establish a benchmark of the system along with responsiveness [94].

Regarding responsiveness, the use of the S6, S8, S10 & S20+ all showed to scale the system according to their screen sizes with no interferences with buttons nor text. This was due to Flutter using SingleChildScrollView [95] a similar tool to Bootstrap [96] from HTML [97] to assist with responsiveness.

The S6, S8 and S10 were used for testing the functionality and ensure compatibility. These proved successful as each device was able to fully run the system with no errors or noticeable delays. The S20+ however, was used for testing functionality as well as testing the speed performance. This performance test will include three tests to get the minimum and maximum speed from each test as shown in Table 10. The measurement of testing is in

seconds (s) which will include the time taken to type and enter fields. The testing was accomplished with an internet speed of 70 Megabits per second. Testing demonstrated that the minimum time to take to perform all tasks with one input (For testing the QR was used as this requires a longer time) took 36.06 seconds. The maximum time taken through testing was 43.83 seconds. Through finding the average mean from each function, an estimate of 39.733 seconds should be had when using CROM.

	Samsung S20+					
	Test 1 (s)	**Test 2 (s)**	**Test 3 (s)**	**Min (s)**	**Max (s)**	**Average (Mean) (s)**
Login	3.92	4.37	3.72	*3.72*	*4.37*	**4.00**
Sign Up	5.90	5.22	6.67	*5.22*	*6.67*	**5.93**
Sign Out	0.91	0.98	0.85	*0.85*	*0.98*	**0.91**
Scan QR	2.59	2.73	2.68	*2.59*	*2.73*	**2.6**
Take 3 Images	10.06	13.12	11.34	*10.06*	*13.12*	**11.50**
Input Data	7.89	8.62	8.44	*7.89*	*8.62*	**8.31**
Submit into DB	0.64	0.76	0.61	*0.61*	*0.76*	**0.67**
Manual Report	0.73	0.51	0.64	*0.51*	*0.73*	**0.62**
Trap Report	0.66	0.71	0.62	*0.62*	*0.71*	**0.66**
All Trap Reports	0.82	0.64	0.69	*0.64*	*0.82*	**0.71**
Trap Map	2.02	1.92	2.06	*1.92*	*2.06*	**2**
Weather	1.68	2.26	1.43	*1.43*	*2.26*	**1.79**
				36.06	*43.83*	**39.73**

Table 10 - Performance Testing for CROM.

4.2 RAW

Testing for RAW was performed on two machines (The main desktop computer and a laptop, broken down in chapter 2.2.1 of the literature review). Testing was also performed on three internet services, Google Chrome, Microsoft Edge and Firefox in terms of performance and capability.

4.2.1 Unit Testing

To maintain consistency, five tests were had for each function to assess stability. The overall success rate was 80% which can be seen in Table 11.

#	Function	Description	Executions	Failures	Success Rate
1	Login	Check DB that there is a match.	5	0	100%
2	Sign Up	Encrypt & check passwords match, ensure email format is used.	5	0	100%
3	Sign Out	User logs out along with resetting cache so no possible re-entry without login/signing up.	5	0	100%
4	Manual Reports	Check entries display in order by connecting from DB, if no entries then display a loading widget with a message.	5	0	100%
5	Trap Reports	Check entries display in order by connecting from DB, if no entries then display a loading widget with a message.	5	0	100%
6	All Trap Data Reports	Check entries display in order by connecting from DB, if no entries then display a loading widget with a message.	5	0	100%
7	Manual Barchart	Select the month to display the bar chart for the current year, further data is shown when clicking on the bar.	5	1	80%
8	Trap Barchart	Similar to the “Manual Barchart” function with the only difference showing information regarding traps	5	1	80%
9	Individual Traps	The latest information of Traps in order along is had with displaying information when clicking on a trap.	5	1	80%
10	Search For Trap Data	Searching is done either through typing or a date picker. Check format is required.	5	3	40%
11	Search for Manual Data	Same as “Search For Trap Data” with the only difference being displaying manual data.	5	2	60%
12	Humidity VS Pest Symptoms	Line Graph displayed with POI to display further information when clicked on for traps.	5	1	80%
13	Insect VS Temperature – Traps	Like “Humidity VS Pest Symptoms” with the only difference being for Insect and Temperature data for traps.	5	1	80%

14	Insect Vs Temperature – Manual	Like “Humidity VS Pest Symptoms” with the only difference being for Insect and Temperature date for manual entries.	5	1	80%
15	Symptoms Per Trap	Ensure the latest trap information is in a pie chart with analysis.	5	2	60%
			75	**13**	**82.66%**

Table 11 - Unit Testing for RAW.

4.2.2 System Testing

Table 12 demonstrates some of the most common use cases that are had in RAW.

Test Case #1	User logins then go to view reports (Manual Report, Trap Reports & All Trap Data Report), once completed the user signs out
Objective(s)	Ensure users can (a) log in by getting credentials from DB, (b) be able to view all reports cleanly and easily, and (c) Users can successfully sign out.
Results(s)	No errors were had during testing. This process mimics and uses the same processes as CROM.
Test Case #2	User logins to navigate to “Statistics”, where they will look at the trap bar chart for a general overview, followed by searching for trap data.
Objective(s)	Ensure users can (a) login successfully, (b) Trap bar chart displays all information regarding traps, and lastly (c), Users can search for traps by a specific date range.
Result(s)	Login and bar chart worked successfully, the only problem that occurred was searching for data in a different format resulted in a crash, this was fixed by applying a check and reformatting the input.
Test Case #3	User logins and navigates to Humidity VS Pest Symptoms which displays the correlation between the two, following this the user also check the correlation of Insect VS Temperature.
Objective(s)	Check that (a) Login is successful, (b) Opening the Humidity VS Pest Symptoms page will display an animated line graph with the correct point mapping along with further insight when a user clicks on a point/dot, (c) Insect VS Temperature should follow the same process as (b) with a slight difference in data.
Result(s)	One minor issue was found; every second data point was skipped due to a loop error. This has been rectified.

Test Case #4	User logins and checks Individual Traps followed by Pest Symptoms Per Trap.
Objective(s)	Check that (a) users can log in with no conflicts, (b) Users can view a bar chart with only recent individual traps information, finally (d) Users can view a pie chart displaying which trap has the worst symptoms with added information.
Result(s)	No errors were found in test case #4.

Table 12 - RAW Test Case.

4.2.3 Performance Testing

RAW proved to be responsive as it was successfully accessible through different monitor screens from 15inch to 36inch monitors. Similar to CROM, the reason for this responsiveness was because of the aid of Flutter along with SingleChildScrollView. As internet services use different resources, a test between Google Chrome, Microsoft Edge and Firefox were had. These tests will include three tests for each internet service to get the minimum and maximum speed from each test. The measurement of testing is in seconds (s) which will include the time taken to type and enter fields. The testing was accomplished with an internet speed of 70 Megabits per second.

Testing from Google Chrome showed that the minimum time to complete all functions of RAW was 49.68 seconds, maximum time of 56.39 seconds. Therefore, the average estimate to complete using RAW in a singular process is 52.93 seconds, as displayed in Table 13.

Testing from Microsoft Edge demonstrated that the minimum time for RAW is 56.16 seconds with a maximum time of 67.69 seconds. The average time estimate for using RAW on Microsoft Edge is 62.14 seconds, shown in Appendix 38. This showing a nearly 10 seconds difference compared to Google Chrome.

Firefox testing presented that the minimum time for RAW is 55.93 seconds with a maximum time of 66.89 seconds. The overall average that it would take a user to use all the functionality of RAW is 61.51 seconds, displayed in Appendix 39. Which is 0.63 seconds different compared to Microsoft Edge. Therefore, the top-performing internet service is Google Chrome. This may be due to Google being the number one backer of Firebase which is hosting the web app.

Google Chrome

	Test 1 (s)	Test 2 (s)	Test 3 (s)	Min (s)	Max (s)	Average (Mean) (s)
Login	10.33	9.89	11.24	*9.89*	*11.24*	**10.48**
Sign Up	17.13	15.67	16.23	*15.67*	*17.13*	**16.34**
Sign Out	0.59	0.62	0.78	*0.59*	*0.78*	**0.66**
Manual Report	1.12	1.06	0.97	*0.97*	*1.12*	**1.05**
Trap Report	1.11	1.04	1.09	*1.04*	*1.11*	**1.09**
All Trap Data	0.92	1.16	1.25	*0.92*	*1.25*	**1.11**
Manual Barchart	2.48	1.86	2.14	*1.86*	*2.48*	**2.16**
Trap Barchart	1.95	2.15	2.27	*1.95*	*2.27*	**2.12**
Individual Traps	1.56	1.42	1.78	*1.42*	*1.78*	**1.58**
Search For Trap Data	5.69	5.09	5.31	*5.09*	*5.69*	**5.36**
Search for Manual Data	4.56	5.12	4.72	*4.56*	*4.72*	**4.8**
Humidity VS Pest Symptoms	1.83	1.45	1.51	*1.45*	*1.83*	**1.59**
Insect VS Temperature - Traps	1.71	1.76	1.84	*1.71*	*1.84*	**1.77**
Insect VS Temperature - Manual	1.87	1.52	1.66	*1.52*	*1.87*	**1.68**
Symptoms Per Trap	1.11	1.28	1.04	*1.04*	*1.28*	**1.14**
				49.68	*56.39*	**52.93**

Table 13 - Google Chrome Performance Testing.

4.3 DOSAN

DOSAN testing provides further testing along with more visualisations as this is the core system. Unit testing & performance testing will still be had but another extension of this testing is dataset testing along with model testing.

Before testing the ML model, an understanding of the behaviour and the characteristics is needed of the data before tackling the design of the model. RNN models require time-series data to be optimal and able to discover trends from a timeline that have regular intervals. This required high-quality datasets since ML models follow the "garbage-in, garbage out" principle.

The reason a high-quality dataset was needed was that not using the correct/inferior data would result in a false positive model. As an example, a healthcare project aimed to treat patients with pneumonia used ML that seemed accurate, but during deployment, it was found that there was missing data regarding asthma (which is highly linked to pneumonia) [98]. This resulted in the model dismissing users with asthma.

High-quality datasets require:

- A sufficient number of high-quality data & labels – Using too little data can result in an improper algorithm trained. The same can be had if there is too much data. There is no recipe to find the perfect fit, but the common issue is that there is too little data. The other aspect is to ensure that the labels are correctly labelled as the ML model will prove ineffective if the wrong labels are used throughout testing and training.
- Balanced dataset – Not to create imbalance, meaning that there should be a similar ratio between positive and negatives values. In this testing, it will be for predicting infestation vs not. As an example, if there are 500 rows of reliable data but only 50 rows of unreliable, then the model will not be able to learn about the unreliable data.
- Consistency – Referring to no missing values or label errors. Meaning that missing values are correctly filled in and that formats are maintained throughout the dataset.
- Relevant – The data should only include details that relate to the overall problem, insect infestation, so including data such as, "Watering crop schedule", is not entirely relevant.
- Diversity – Ensuring there are no bias factors that could affect the model, thus focusing on multiple aspects rather than just the end goal [99]. For example, a self-driving car is trained with only factors that are had on a highway, then it will be unusable in the city.

Due to COVID-19 it made it very difficult to gather the required data for research/testing purposes as one of the issues surrounding ML is data, data may be scarce or scattered making it rather difficult to gather and compile efficiently without being a lengthy process [100]. Therefore, knowledge gained from the papers in the literature review in section 2 was used. This knowledge being was to create synthetic data while following the general understanding of pest infestation factors with the help of an entomologist.

The dataset that is being used for testing is done through XLSX format as the majority of datasets can be converted into an XLSX. The synthetic data created followed a realistic timeline which has 4474 rows of data, while not being biased and allowing a trend to take place. The data started on 01/06/2017 and ended on 30/06/2021 (4 years). The reason for creating this period is that training a model on a specific section of data that has high values should be avoided. Instead incorporating all aspects from low to high will allow the model to learn and be able to give significantly more accurate predictions.

This dataset had the following columns which are displayed in Table 14.

Column Name	Description
Date	Continuous date from 01/06/2017 - 30/06/2021.
Insect	Three insects were used, Melon Fly, Cotton Bollworm and Mango Fruit Fly
Crop	Three crops were chosen concerning insects preferred choice: -Tomato = Melon Fly, -Maize = Cotton Bollworm, -Mango = Mango Fruit Fly.
Temperature	The temperature was categorised in: - "C" for cold (0° to 9°), - "L" for low (10° to 15°), - "A" for average (16° to 24°), - "H" for high (25+°).
Number of Insects	The number of insects that have been recorded.
Season	Tracking the season of the year; spring, summer, autumn, winter.
Humidity	Humidity is categorised into: - 0 = humidity < 69%, - 1 = humidity >70%.
Location	The country/county where the data is had, for this test, Buckingham and Milton Keynes were chosen.

Crop Habitat	Were crops are stored, this follows: - Tomato = Glasshouse, - Maize = Glasshouse, - Mango = Open field.
Wind	Wind is categorised into the following: - Low = Wind < 18 mph - High = Wind >19 mph
Rain	Rain is also categorised: - High = Precipitation rate > 5 mm per hour. - Low = Precipitation rate < 4 mm per hour.
Sunlight	The number of hours of sunlight there was for the day.
Infestation	Categorised as binary: - 0 for no, - 1 for yes.

Table 14 - Dataset Creation Pattern.

While the dataset may have some undiscovered patterns, it was still important to implement a few patterns throughout the dataset to ensure the ML model will find them.

The pattern for certain fields is:

- When the wind is:
  - *low,* then humidity = 1.
  - *high*, then humidity = 0.

- When temperature is:
  - *Cold*:
    - Number of insects = 0-10.
    - Sunlight between 0-6 hours.
  - *Low*:
    - Number of insects = 9-24.
    - Sunlight between 7-14 hours.
  - *Average*:
    - Number of insects = 15-45.

        - Sunlight between 7-14 hours.
    - *High*:
        - Number of insects =0-8
        - Sunlight between 7-14 hours.

- When there is high wind & high rain then the number of insects = 0 -10.

This pattern greatly assisted in building the dataset with a functional flow. I highly recommend that developers that are needing to use synthetic data for the ML model follow a similar methodology. Otherwise, this can delay delivery times as further post-processing will be needed with a higher risk of the ML model not being able to train correctly.

Furthermore, just creating the dataset is data analysis. Data analysis is vital to ensure the data is suitable for the model to learn on as well as highlighting potential areas for further processing. The first set of analyses is ensuring that a trend is found as if no trend is visual then the dataset will need further processing. The dataset has a repeating trend with an equal distance between points which is visually represented in Figure 34.

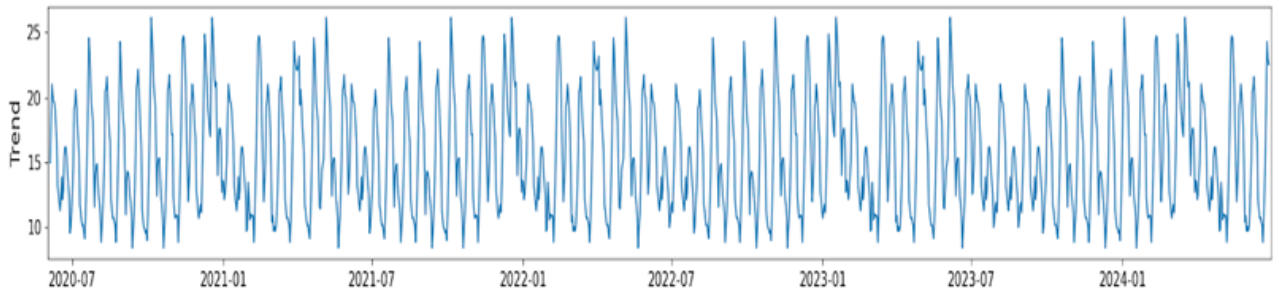


Figure 34 - DOSAN dataset trend occurrences.

Diving deeper into the created dataset, analysis of data in the form of a scatterplot occurs. This is to ensure it is usable or not by creating functions to display relationships that are had between all columns. The scatterplot also demonstrates how much one variable is affected by another. For example, Wind (X-axis) vs Number of Insects (Y-Axis) demonstrates a general positive correlation as the wind increases the number of insects decreases.

Final dataset testing involves utilising the Dickey-Fuller test (DF), which is a statistical analysis to determine if the dataset is (a) reinforcing that the dataset is time series and (b) if

the time series is stationary or not. To understand what stationary is, we must first look at what time series requires certain characteristics:

a) Trend – the movements between high and low values over a period.

b) Seasonality - referring to the repeating pattern that is had over a fixed period.

c) Irregularity - referred to as noise, meaning that not all data will be a unified pattern and that there are outliers with short durations of irregularity.

d) Cyclicity - similar to seasonality but the duration is unfixed and the space between two cycles can be longer.

If trend or seasonality are had then it means the time series is not stationary [101]. Typically, stationary datasets are easier to model but in general, will have no predictable patterns in the long term for the ML model to predict [102]. However, we are looking to use the ML model to predict values that are reliant on seasons and trends. DOSAN includes the use of a decomposition, graphical representation of the DF, as illustrated in figure 35. The decomposition graph includes several graphs within itself. The first graph is the overall trend for the dataset. The second graph is the mean average of the trend to demonstrate in a simpler view. The third graph is the graph is demonstrating the seasonality that is had throughout the four years' worth of data. Trend & Seasonality graphs highlight the visibility that the time series in use is non-stationary. The last graph refers to residual which is the difference between the scatter plot point and what a regression equation predicts. This motivates the use of ML to reduce outliers.

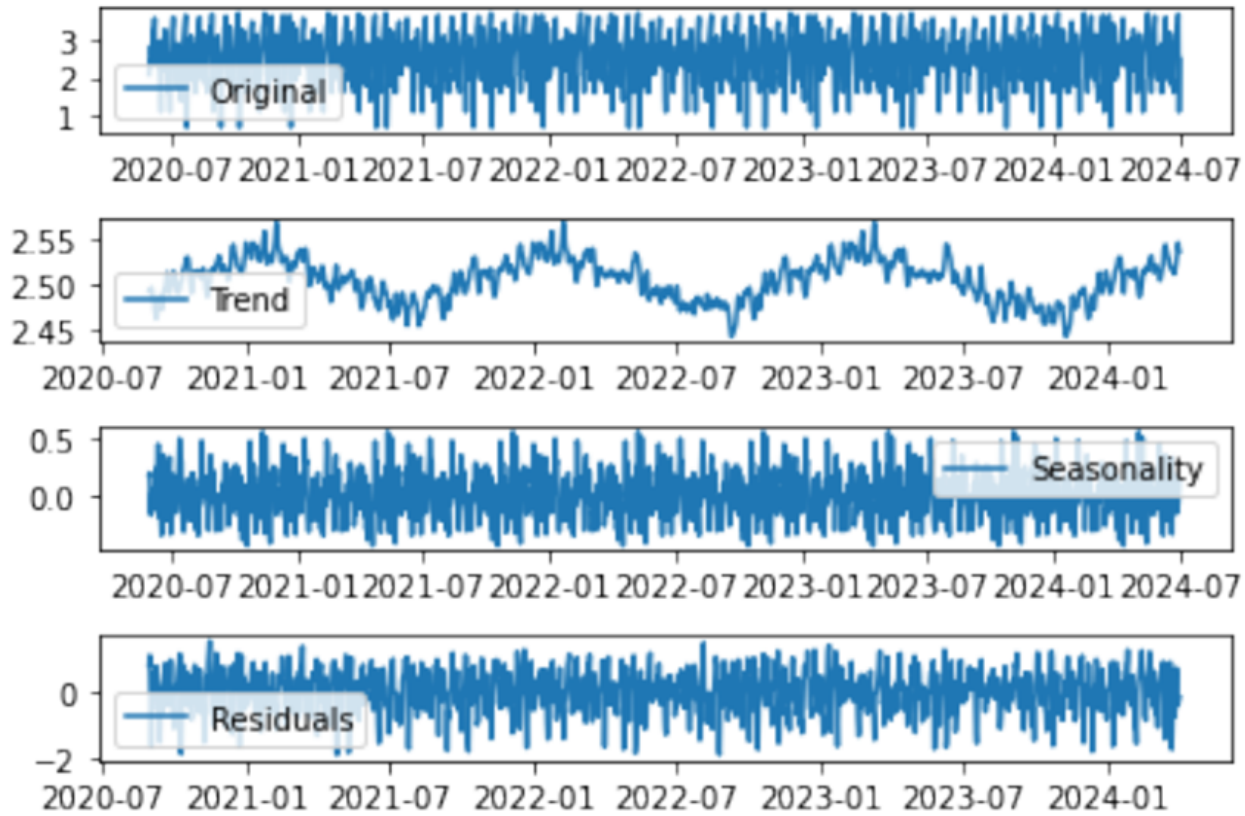


Figure 35 - Decomposition Graph for DOSAN Testing Dataset.

The dataset has now been proven to be usable for DOSAN model training. However, there was just one more stage for post-processing. This is converting the data into numeric fields rather than have a mixture of strings and integers. While this may seem unnecessary, there have been reports that show that using numerical data is faster for ML model training [103]. This will be tested further along in the report to assess the validity of reports. The data has been transformed to fit the following numerical constraints:

- Crop:
    - Tomato = 0,
    - Maize = 1,
    - Mango = 2.
- Insect:
    - Melon Fly = 1,
    - Cotton Bollworm = 2,
    - Mango Fruit Fly = 3.
- Temperature:
    - “C” = 0,

    - “L” = 1,
    - “A” = 2,
    - “H” = 3.
- Location:
    - Buckingham = 0,
    - Milton Keynes = 1.
- Wind:
    - Low = 0,
    - High = 1.
- Rain:
    - Low = 0,
    - High = 1.
- Crop Habitat:
    - Glasshouse = 0,
    - Open field = 1.

The important thing is to scale features before training the neural network. A common way of doing this is through standardization scaling, by subtracting from the mean and dividing by the standard deviation of each feature for improved performance from the model and accuracy. One way of achieving this is to use the `tf.keras.utils.normalize` [104] technique that rescales the values into a range of [0,1]. For this test though, the use of manual standardisation, displayed in Figure 36, demonstrates how simple it is achieved. As a note, it is recommended to use the train split for the mean and standard deviation as this is what the model will be using to learn.

```
[ ]  TRAIN_SPLIT = 1200 # 80%
     dataset = features.values
     data_mean = dataset[:TRAIN_SPLIT].mean(axis=0)
     data_std = dataset[:TRAIN_SPLIT].std(axis=0)
```

```
[ ]  dataset = (dataset-data_mean)/data_std
```

Figure 36 - Scaling data for ML model.

The model used for testing is an RNN LSTM model that was decided from chapter 2 in the background research. RNN's can predict either only one day specified (normally the next day) or can predict multiple future days. We did not want to limit DOSAN testing to only one of these methods, Univariate and Multivariate model testing is had. The RNN models for multivariate and univariate will be using the same parameters choices from the background research in chapter 2 as this showed to be the optimal parameters for model accuracy. Testing will not include any models that are overfitting or underfitting.

4.3.1 Multivariate

Several steps were needed in deciding the best model for final production based on the dataset and a future target of 50 days. Through the help of DOSAN GUI, this was done with ease.

1. Step one was deciding on the best train/test split. After vigorous testing with all parameters set as default values expect only changing the train and test split percentage. From testing, the best train/test split was 70%/15%. Displayed in Appendix 40 is a table of testing train/test splits.
2. Step two, deciding values for seed & batch. Taking the train/test split findings into consideration, testing showed that a seed of 13 and a batch of 256 was best. Refer to Appendix 41 for the table holding all findings concerning seed & batch.
3. Step three including looking at the buffer size to be used by taking steps one & two into account. Testing showed that a buffer size of 1000 is best used. Appendix 42 table demonstrates all testing involved in buffer size decisions.
4. Step 4, internal evaluation decision. Testing proved that 250 is best used. Referring to Appendix 43 table highlights the other values that were tested.
5. Step 5, comparing different step values, through testing the best step shown was a step of 2. Appendix 44 table includes all testing for step decision.
6. Step 6 looks at what values to use for LSTM 1 & LSTM 2. Proven values show that LSTM 1 is 128 with LSTM being 256. A table highlighting all testing is in Appendix 45.
7. Step 7 lastly is deciding on the validation steps. The most optimal validation step found for this dataset was 100. Further details are shown in the table in Appendix 46.

Therefore, in summary from testing, it was proven that the best multivariate model will be the parameters along with the loss being 0.1125 and the validation loss (val_loss) as 0.1062, as shown in Figure 37.

Seed	Batch	Buffer	Evaluation Internal	Train split	Past History	future Target	Step	Epoch	LSTM 1	LSTM 2	Validation steps	Loss	Val loss
13	256	1000	250	(70%)	(15%)	50	2	10	128	256	100	0.1125	0.1062

Figure 37 - DOSAN multivariate parameters.

The loss and val_loss showed that this model is learning very well as there is no overfitting nor underfitting. Overfitting is when training loss is less than validation loss and means that the model fits the noise from the data. Often overfitting results in a low error on the training set but a high error on test/validation sets [105], Underfitting is when training loss is greater than validation loss, this meaning that the model does not capture the underlying trend of the data and often caused from a result of a simple model used. This is noticeable not only by the values being close to one another but visually too. Typically, a loss and val_loss graph are used when training ML models for diagnosing. The multivariate RNN model as displayed in Figure 28 demonstrates a good fit as there is a little gap between the two and decreases over time together [106].

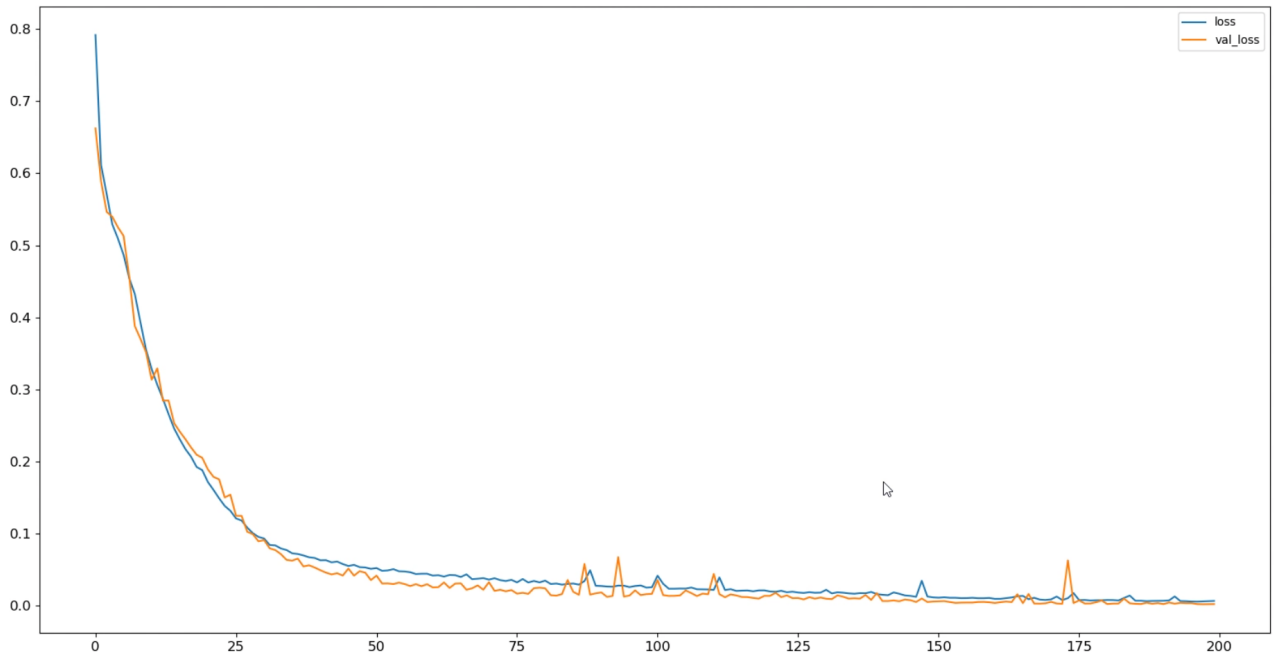


Figure 38 - Loss graph for the multivariate model.

The final step of validation is ensuring the model is accurate. A graph displaying fifty points were used as future targets (demonstrated in blue) with previous trend data to the left. Using the trained multivariate model was used to try predicting the “unseen” future targets (demonstrated in red). As shown in Figure 39, the model was able to closely predict the future values, thus proving successful model generation based on the dataset.

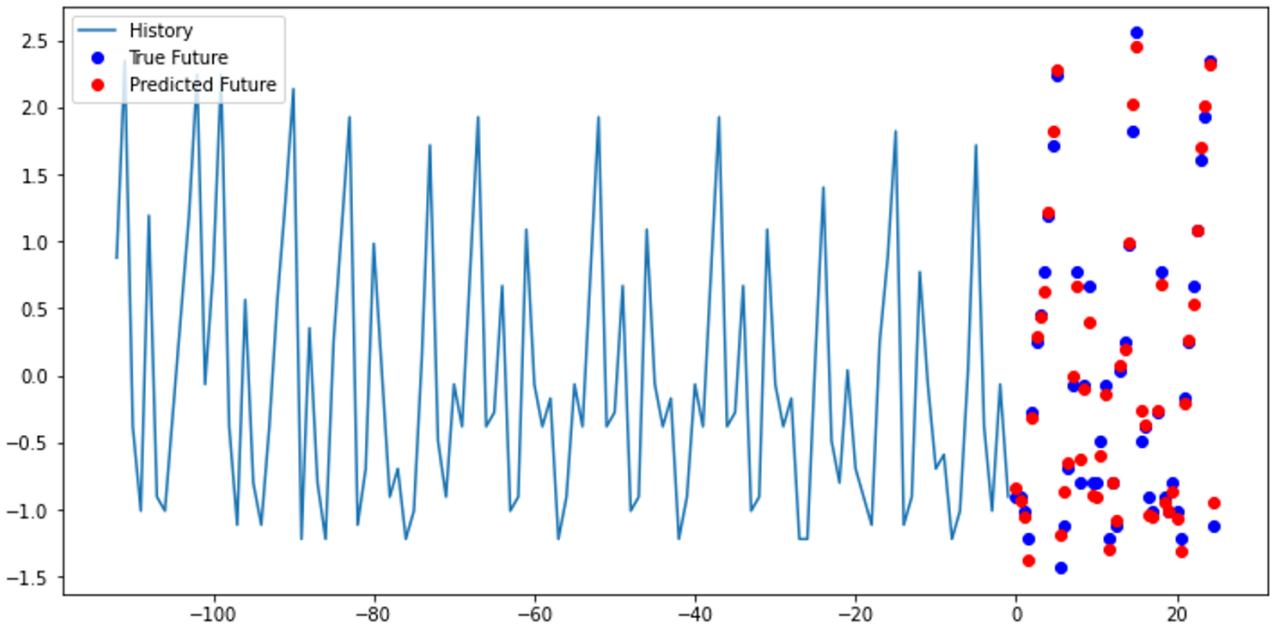


Figure 39 – Prediction graph for the multivariate model.

Before continuing onto the univariate model testing of DOSAN, a few points to mention that were found through testing.

1. Increasing the history/test split takes the model longer to train.
2. Increasing the seed makes training slower.
3. An increase in internal evaluation also slows the training.
4. Lower steps increase training time.
5. The use of the LSTM greatly reduces the loss, however, the use of too many LSTM's will create errors/negative outcomes.
6. An increase in LSTM values will create a slight delay in training.
7. The more validation steps used, the slower the training.

4.3.2 Univariate

The univariate model testing is using the same created dataset. Univariate model generation and testing followed a similar process of multivariate testing. Once again, this process of testing was done with ease thanks to DOSAN GUI. The future target for testing is 0, which entails that it is predicting the next day. The steps for testing the univariate model followed:

1. Step one - deciding on the best train/test split. After testing with parameters set as default values, the best train/test split was 80%/15%. Displayed in Appendix 47 is a table of testing train/test splits.

2. Step two - deciding the values for seed & batch. Testing showed that a seed of 4 and a batch of 32 was best. Refer to Appendix 38 for the table holding all findings of seed & batch.
3. Step three – deciding on the buffer size to be used by taking steps one & two into account. Testing showed that a buffer size of 5000 is best used. Appendix 49 table demonstrates all testing involved to buffer.
4. Step 4- deciding on the internal evaluation decision. Testing proved that 200 was best. Referring to Appendix 50 table highlights the values through testing.
5. Step 5 – looking at the number of epochs, testing showed that the best is 200 epochs. Appendix 51 table includes all testing for epoch decision.
6. Step 6 – deciding the number of validation steps that are best used for the dataset used. Which was found that 50 validation steps are best. A table highlighting all testing is in Appendix 52.
7. Step 7 lastly is deciding on the LSTM. The best LSTM value found for this dataset was 16. Further details are shown in the table in Appendix 53.

Testing proved that the univariate model for this dataset has a loss of 0.0049 and a validation loss of 0.0021 as shown in Figure 40.

Seed	Batch	Buffer	Evaluation Internal	Train split	Past History	future Target	Epoch	Validation steps	LSTM	Loss	Val loss
4	32	5000	200	1193 (80%)	223(15%)	0	200	50	16	0.0049	0.0021

Figure 40 - DOSAN univariate parameters.

The univariate RNN model as displayed in Figure 41 demonstrates a good fit as there is a little gap between the two and decreases over time together. However, this model could still be improved as there were certain sections where the model got confused, this can be seen via the spikes in the loss graph but then went back on track.

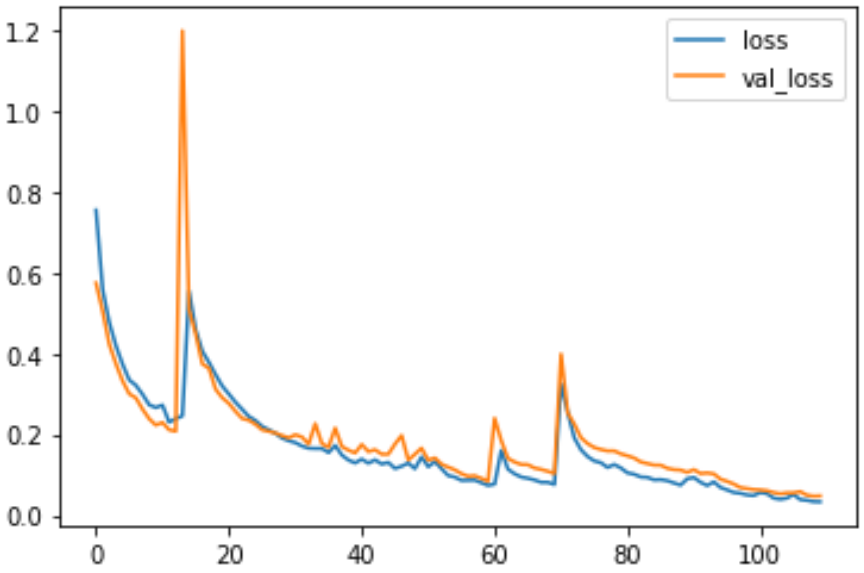


Figure 41 - Loss graph for the univariate model.

To illustrate the importance of ML, a feature of DOSAN is to demonstrate a baseline prediction which is using standard mathematics to forecast. However, as shown in Figure 42, this is far from the actual true target. Using ML with a next day target. A graph displaying a next future value point was used (demonstrated in red) with previous trend data to the left of this. Using the trained univariate model to try predicting the "unseen" future targets (demonstrated in green). As shown in Figure 43, the model was predicting exactly the next day's values.

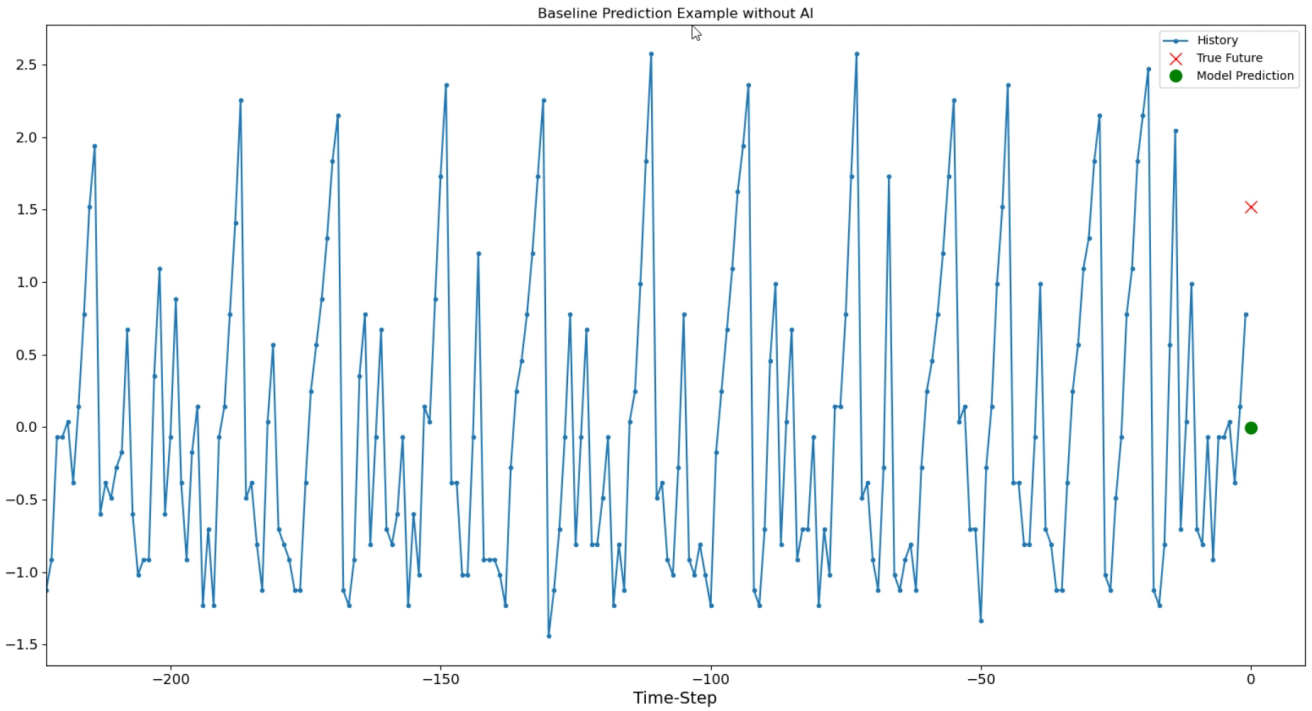


Figure 42 - Baseline Predicting for Univariate DOSAN.

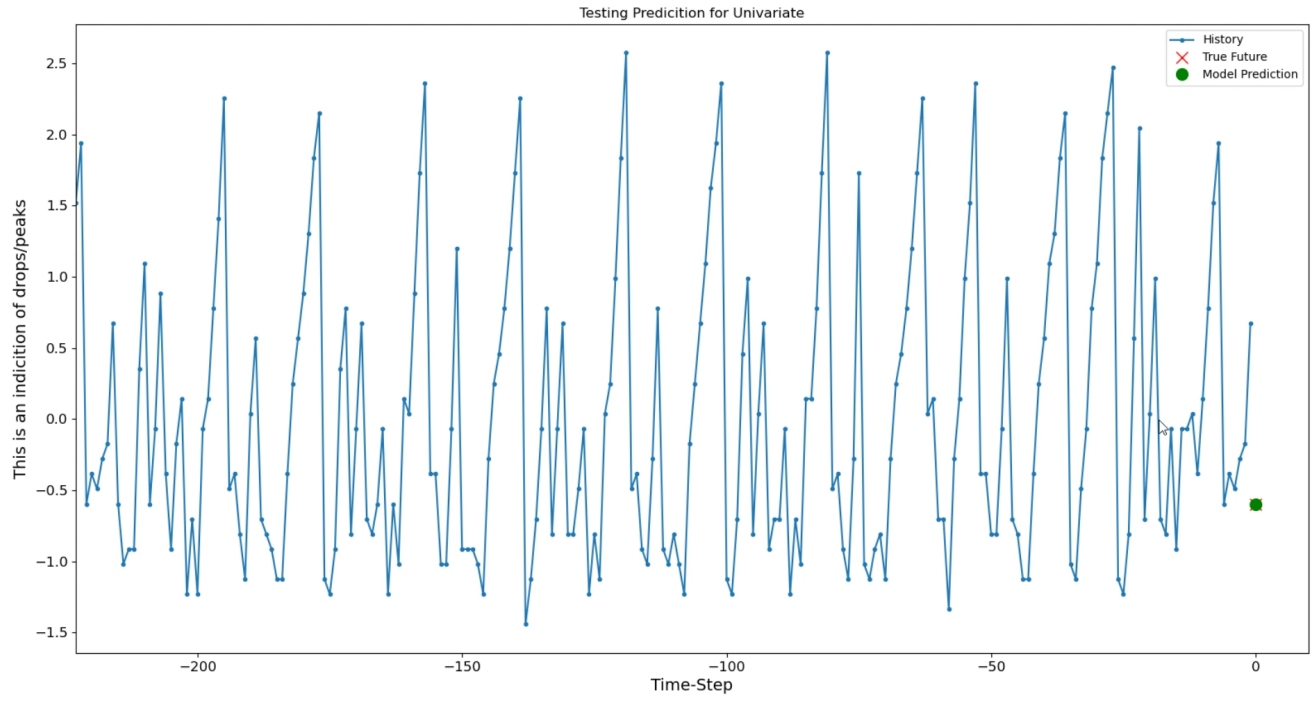


Figure 43 - Univariate Prediction for DOSAN.

Now that multivariate and univariate model testing has been accomplished, the next stages are unit testing, performance testing and further discussion testing topics.

4.3.3 Unit testing

The unit testing for DOSAN is like CROM & RAW and looks at the whole system, not just the ML model generation. To maintain consistency, five tests were had for each function. The overall unit testing success rate was 79% as displayed in Table 15.

#	Function	Description	Executions	Failures	Success Rate
1	Dataset Explorer	Ensure that file explorer only displays files in the correct extensions.	5	0	100%
2	Columns to Use	Check button generation expands with all columns found in the datasets.	5	1	80%
3	Save Compiled Dataset	The newly compiled dataset can save	5	1	80%
4	Univariate – Select Index	Ensure only columns that are continuous data are able for selection	5	0	100%
5	Univariate Select One Target	Ensure all other columns are able for selection and the only one can be used.	5	0	100%
6	Univariate Train Split	Check that the percentage button choice is saved correctly.	5	0	100%
7	Univariate Test Split	Check that the percentage buttons only display choices that are in correlation to Train Split choice. This too needs to be saved correctly.	5	1	80%
8	Univariate Parameters	Check all dropdowns are working in correlation to parameters choices as well as being saved to be passed into model training.	5	2	60%
9	Univariate Model Training	Ensure that the model can train regardless of entries of parameters.	5	4	20%
10	Multivariate Select Index	Ensure only columns that are continuous data are able for selection	5	0	100%
11	Univariate Select Multiple Target	Ensure all other columns are able for selection.	5	0	100%
12	Multivariate Train Split	Check that the percentage button choice is saved correctly.	5	0	100%
13	Multivariate Test Split	Check that the percentage buttons only display choices that are in correlation to Train Split choice.	5	0	100%

14	Multivariate Parameters	Ensure all parameters are saved correctly and saved in the correct format	5	0	100%
15	Multivariate Model Training	Ensure that the model can train without crashing.	5	3	40%
16	Download Model	Check that the trained model can be saved in the correct format for future implementation.	5	1	80%
17	Data Analysis	Should display a histogram from all data in the dataset.	5	1	80%
18	Loss Graph	Ensure the loss graph from model training is displayed	5	0	100%
19	Root Mean Graph	Check that the root mean graph is displayed from model training	5	0	100%
20	Predictions	The function should display predictions with past data, true future values, and model predictions.	5	3	40%
21	Baseline Predictions	Ensure that a graph is shown with a prediction without the assist of ML.	5	2	60%
22	Scatterplot	Check that a correct scatterplot is being displayed with all data from the dataset compiled.	5	2	60%
23	Dickey-Fuller test	A full DF graph test is displayed from user-selected columns to the dataset.	5	3	40%
			115	24	79%

Table 15 - Unit Testing for DOSAN.

4.3.4 Performance testing

DOSAN proved to be responsive as it was able to run on both a laptop with a subpar specification machine and a high-end specification machine, as detailed in the literature review, section 2.2.1. Similar to previous performance testing which that three tests will be had. The testing will be focusing on a multivariate model as this training often takes the longest and requires the most processing.

Testing from the laptop machine showed that the minimum time to complete all functions for DOSAN is 2081.97 seconds, maximum time of 2975.64 seconds. Therefore, the average estimate to complete using DOSAN in a singular process is 2529.49 seconds, as displayed in Table 16.

Testing from the desktop machine showed that the minimum time to as 1470.17 seconds, a maximum time of 1750.58 seconds. Therefore an average estimated time to complete using DOSAN in a singular process is 1591.21 seconds, as displayed in Table 17. The difference between using the laptop and desktop is 938.28 seconds in favour of the desktop.

	Laptop					
	Test 1 (s)	Test 2 (s)	Test 3 (s)	Min (s)	Max (s)	Average (Mean) (s)
Dataset Explorer	4.12	3.79	3.86	*3.79*	*4.12*	**3.92**
Columns to Use	6.28	6.81	7.04	*6.28*	*7.04*	**6.71**
Save Compiled Dataset	5.31	5.12	4.60	*4.60*	*5.31*	**5.01**
Index Selection	3.01	3.19	2.93	*2.93*	*3.19*	**3.04**
Target Selection	3.57	4.31	3.62	*3.57*	*4.31*	**3.83**
Train Split	3.03	2.87	2.94	*2.87*	*3.03*	**2.94**
Test Split	3.37	3.27	2.81	*2.81*	*3.37*	**3.15**
Parameters	8.05	9.24	8.63	*8.05*	*9.24*	**8.64**
Model Training	2463.21	2903.56	2020.05	*2020.05*	*2903.56*	**2462.27**
Download Model	9.39	8.10	9.22	*8.10*	*9.39*	**8.9**
Data Analysis	1.90	2.14	2.05	*1.90*	*2.14*	**2.03**
Loss Graph	1.50	1.89	1.28	*1.28*	*1.89*	**1.55**
Root Mean Graph	1.06	1.22	1.69	*1.06*	*1.69*	**1.32**
Predictions	1.95	1.11	1.55	*1.11*	*1.95*	**1.53**
Baseline Predictions	1.86	1.99	1.35	*1.35*	*1.99*	**1.73**
Scatterplot	2.57	3.34	3.39	*2.57*	*3.39*	**3.1**
Dickey-Fuller test	9.65	10.03	9.79	*9.65*	*10.03*	**9.82**
				2081.97	*2975.64*	**2529.49**

Table 16 - Performance Testing for DOSAN Using Laptop.

Desktop

	Test 1 (s)	Test 2 (s)	Test 3 (s)	Min (s)	Max (s)	Average (Mean) (s)
Dataset Explorer	3.01	3.40	3.31	*3.01*	*3.40*	**3.24**
Columns to Use	6.77	6.75	5.74	*5.74*	*6.77*	**6.42**
Save Compiled Dataset	4.63	4.72	4.48	*4.48*	*4.72*	**4.61**
Index Selection	2.61	3.27	3.15	*2.61*	*3.27*	**3.01**
Target Selection	3.26	2.93	3.17	*2.93*	*3.26*	**3.12**
Train Split	3.09	3.23	2.99	*2.99*	*3.23*	**3.10**
Test Split	3.25	2.78	3.08	*2.78*	*3.25*	**3.03**
Parameters	9.49	8.07	8.89	*8.07*	*9.49*	**8.81**
Model Training	1414.08	1686.50	1491.53	*1414.08*	*1686.50*	**1530.70**
Download Model	8.08	8.17	7.86	*7.86*	*8.17*	**8.03**
Data Analysis	1.27	1.57	1.50	*1.27*	*1.57*	**1.44**
Loss Graph	1.22	1.36	0.72	*0.72*	*1.36*	**1.1**
Root Mean Graph	1.31	1.27	1.15	*1.15*	*1.31*	**1.24**
Predictions	1.06	1.14	1.68	*1.06*	*1.68*	**1.29**
Baseline Predictions	1.79	1.95	1.06	*1.06*	*1.95*	**1.6**
Scatterplot	1.71	1.69	1.83	*1.69*	*1.83*	**1.74**
Dickey-Fuller test	8.72	8.67	8.82	*8.67*	*8.82*	**8.73**
				1470.17	*1750.58*	**1591.21**

Table 17 - Performance Testing forDOSAN Using Desktop.

4.3.5 Further Testing

Even though the multivariate and univariate models have been proven to work. There were a few other tests that were had to assess different types of datasets and their outcomes.

One of the tests looked at what would occur if there was no continuous data used for indexing, thus making the time series data stationary. The way of going about this was to randomly delete gaps in the dataset. Through testing it was found that the model was not able to train correctly and was getting confused throughout the process, this is evident in the loss & val_loss graph in Figure 44. The final model prediction shows that the model was not able to correctly forecast, as shown in Figure 45 and Figure 46.

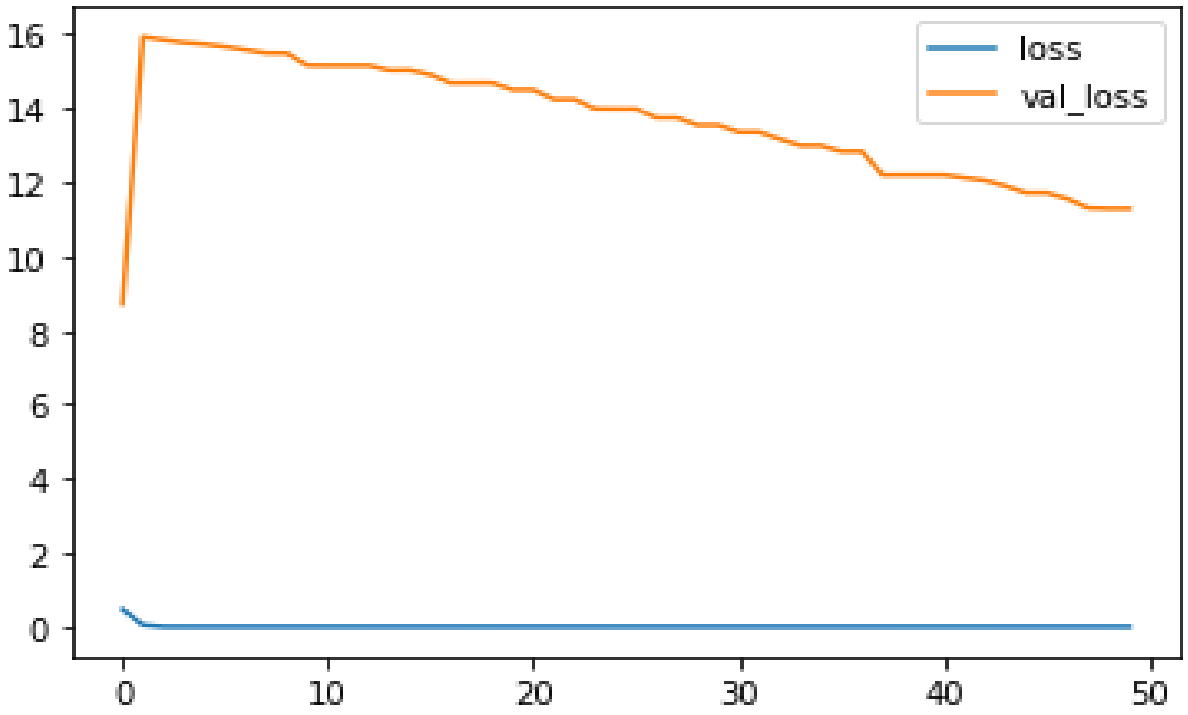


Figure 44 - Loss Graph for non-continuous training.

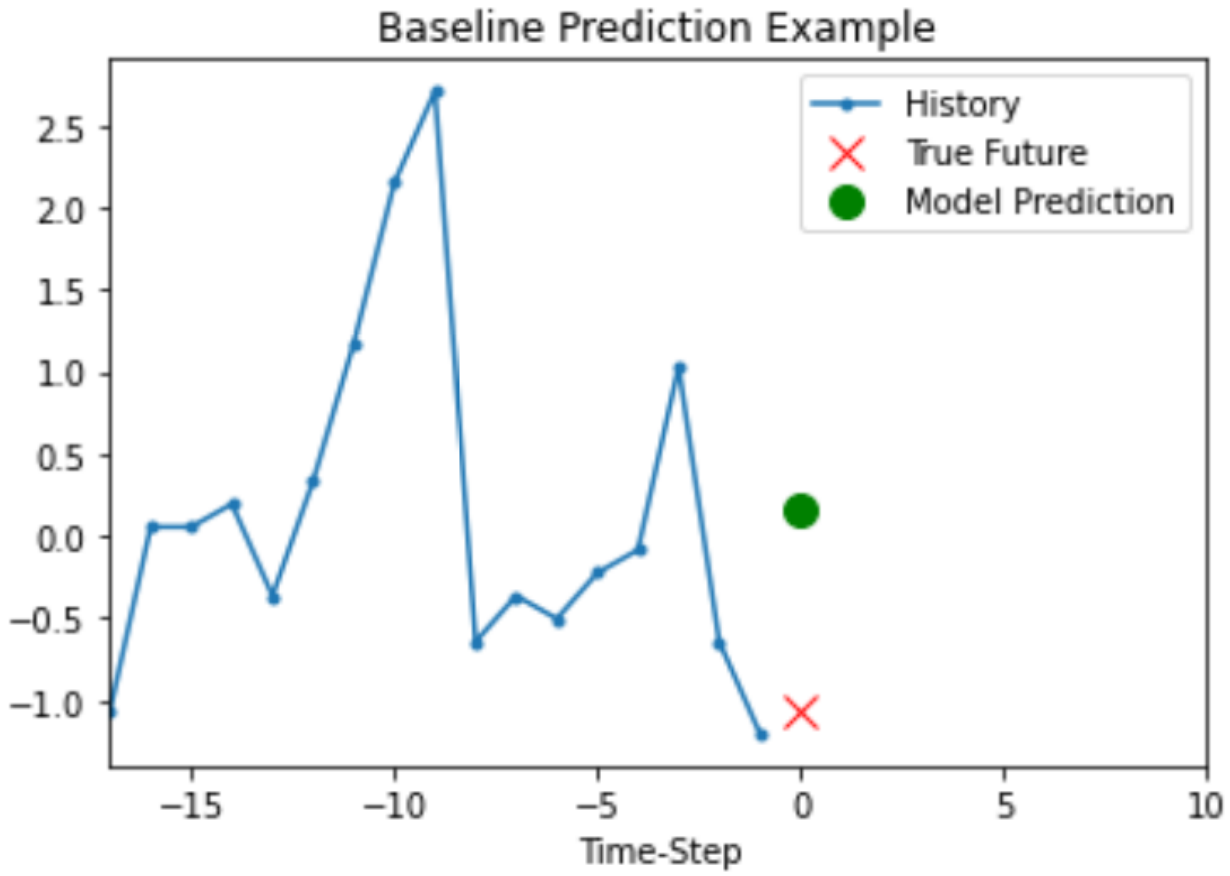


Figure 45 - Non-continuous data for univariate.

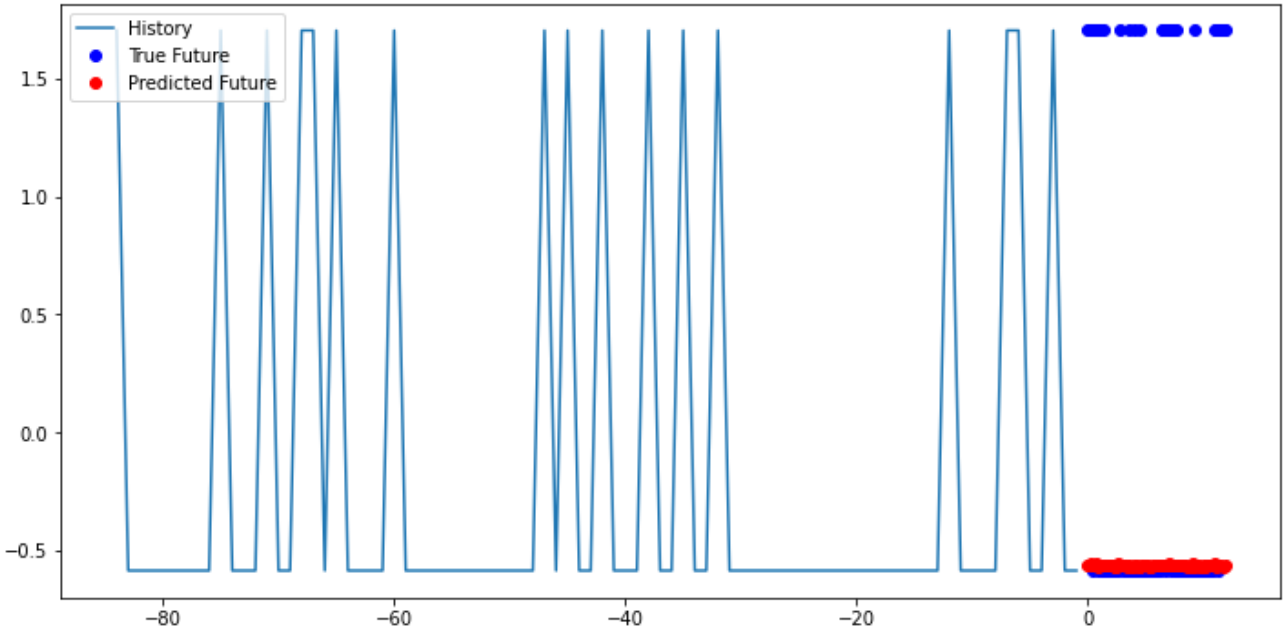


Figure 46 - Non-continuous for multivariate.

One other further test was an evaluation of data that is in a text format vs numerical format for performance comparison. The model used the same parameters that were tested upon

earlier for the univariate model. The model was able to exactly predict the future day, as shown in Figure 47, however, the only difference that was found was that the original text dataset before numerical processing, took 20+ minutes longer.

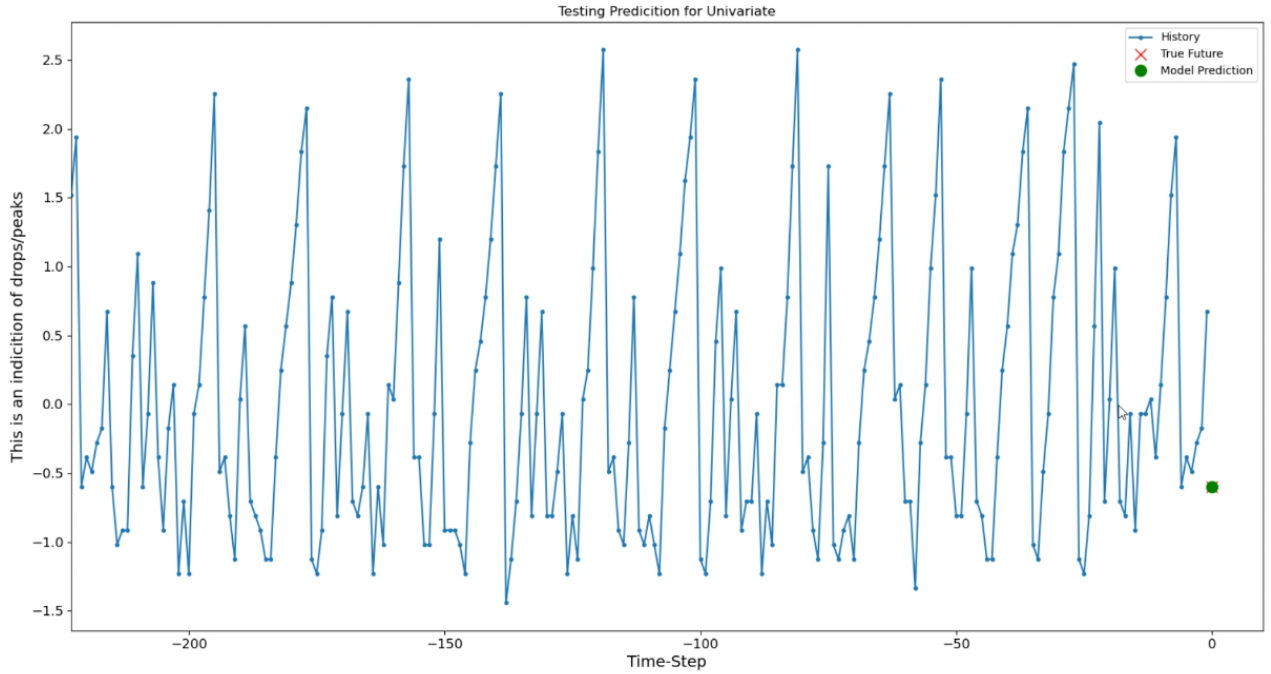


Figure 47 - String model testing for DOSAN.

Assessing further tests was had to ensure that DOSAN works will other datasets along with the time it would do a manual method. These datasets along with outcomes are:

1. One scenario included three datasets (Location, Insect and Weather) were used which equated to around 2000 rows worth of data. The datasets included:

    a. Location included the following columns with relevant data, Climate, Date, Sea Level, Country, and Town.

    b. Insect dataset included Insect Type, Infestation Growth Rate and Date.

    c. Lastly was the Weather dataset that included Humidity, Temperature, Region, and Date.

    The purpose of this test was to build a model that can be used to find the correlation between Insect Type, Sea Level and Humidity. The testing used all the default parameters set by DOSAN and a test split of 80% for training and 20% for testing. The outcome was that the model was able to successfully train, however, further modifying of parameters should be had to make the model even better.

2. Another scenario included two datasets, Person and Heart, with the idea to create a model to look at the trend between resting blood pressure, age, sex, and cholesterol. These datasets included:

    a. Person dataset which has Age, Sex, and Resting Blood Pressure.

    b. The heart dataset includes Resting Blood Pressure, Cholesterol, and Chest Pain Type.

    The result was that the model was able to train but not very successfully based on that the dataset only had 1000 rows in total and used a similar train/split to scenario 1. However, if a train/split of 60/40 was used then this may improve the accuracy as well as further tuning of the parameters.

3. One last scenario included only one dataset, Air Quality, of 1500 rows which included the following columns which are sensors for metal oxide:

    a. Air Quality dataset includes Date, Time, CO, PT08.S1, NMHC, C6H6, PT08.S2, NOx, PT08.S3, NO2, PT08.S4, PT08.S5, Temperature, Relative Humidity and AH Absolute Humidity.

    The objective of this was to create a model to find the correlation between the relative humidity and the PT08 sensors. The result was that the model was able to train well considering default values for parameters were used and a train/test split of 80/20, this can be seen in Figure 48 where the model was able to mostly predict the true future values with slight outliers.

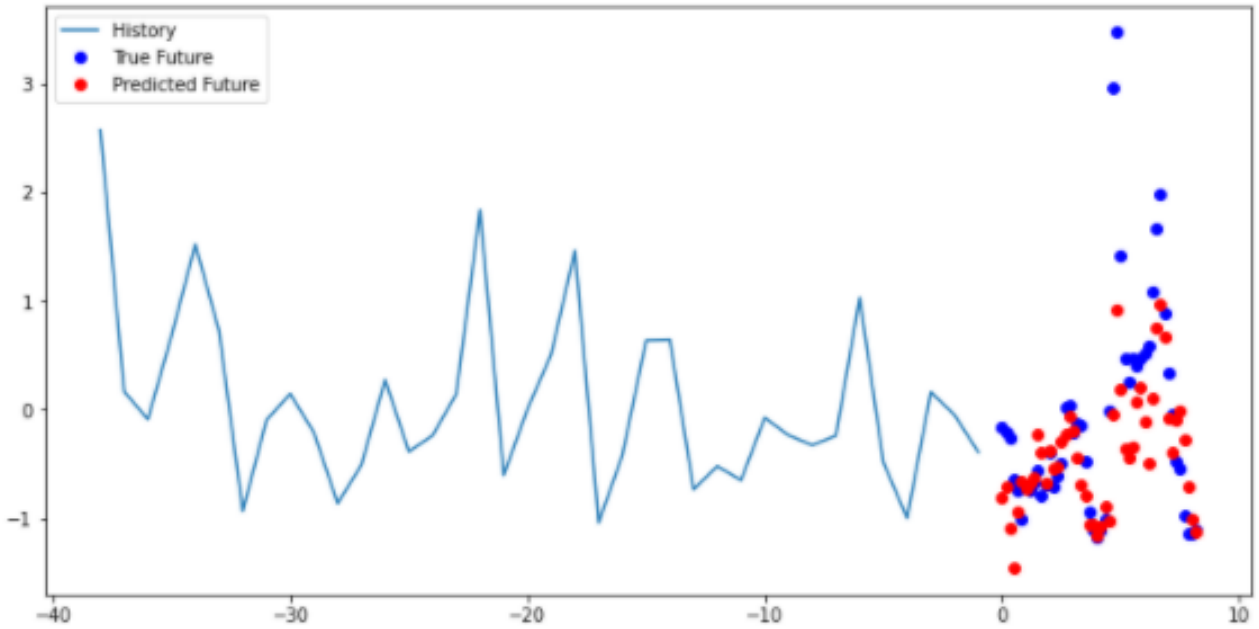


Figure 48 - Air Quality DOSAN Prediction.

As a comparison to the performance of DOSAN, a manual model was had. The manual ML model from Google Colaboratory took two weeks to build one dataset, and then another hour for training it. Further development took the analysis section similar to DOSAN roughly one week. Therefore, in total, took 3 weeks to build a model with analysis were as DOSAN can do all of this in less than an hour. The reason the manual analysis took longer is due to (a) searching for existing systems that are compatible with model and data analysis, (b) no method was compatible and all required training; so, a manual method was created based on any/each specific data.

This testing proved effective in evaluating DOSAN as a whole and highlights the methodology that is had internally. DOSAN is not limited to only insect prediction but can be used in a wide variety of scenarios.

Chapter 5 Conclusion & Future work

Prediction is one of the key factors that ML is known to assist in, however, this requires lengthy manual pre-processing. There are numerous papers and applications out there that discuss and implement ML that achieves an accuracy of 80%+, but the common pattern that was found in this research is that data was already cleaned and collected along with a model structured to work solely for specific data. While this technology is still growing and ML has come a long way in terms of performance, it still does not meet the industry as a tool that can be easily acquired and utilised. This is why this research looks at a rather different perspective of the use of ML and integration into real-world scenarios.

I feel my approach to this research was rather a waterfall which did work well for some activities however, an agile approach would have been more beneficial due to constant changes that were occurring. Using the agile approach also allows for faster delivery. The other reason is that agile seems to be a growing standard in the industry.

While this system is aimed at aspects such as insect prediction along with farming applications, it does not get limited to only these fields. Slight UI changes can make this accessible for a wide range of other industry applications such as stocks & shares, electricity consumption, etc., making this research a framework for other areas of work.

Future improvement & development of CROM would be to utilise TensorFlow Lite [107], adding the functionality of ML for image recognition of the insects that are being captured via the camera. A future improvement for RAW would be to link the model analysis that was trained through DOSAN into a similar format as the other reports. For example, there could be a tabular form card that keeps records of all previous model training data such as what parameters were used etc. This allowing for analysts to reimplement their training easier by referring to previous tests.

The future improvement of DOSAN would be to incorporate image recognition software that can build labels from given datasets and then be trained to give predictions on the images that have been inputted. One method of achieving this additional functionality would be to incorporate a menu to choose between an RNN or a CNN model prediction application. Once the labels are predicted for the images they can be saved into an external dataset, this then linking to other datasets for an RNN prediction. The other enhancement to DOSAN can be to implement the system in a cloud-based platform that will make it more robust as well as platform and hardware independent. Cloud hosting can be achieved using AWS (Amazon Web Services) for easy deployment and maintenance.

The use of CROM, RAW and DOSAN order allows for fast processing of data along with an ML model that can be used in any industry with a few small alterations.

5.1 Summary of achievements:

- Generation of data along with high-level analysis is often a separate task involving different pre-processing and software. CROM tackles the issue of data generation through an easy-to-use mobile application that anyone can use along with analysis on the go. Followed using CROM is RAW, which extends the level of analysis and dives deeper to provide significantly more detail in terms of the data recorded in a uniform way consistently. CROM & RAW were able to successfully input data within one minute including storage into the cloud database.

- Prediction is an area that can be further utilised in technologies as it allows for further insights into hidden trends and possible solutions for problems. DOSAN has allowed the use of building an RNN LSTM model based on any data presented so long as there is a foreign key that can be found. DOSAN is not just used for the initial building of a model but can be used for updating a model that is implemented in an application. For example, if there is another data point added to an existing model then this will result in an error as the model has not seen this data point and thus will ignore it, but retraining the model with the added data point will be a simple step of just reimplementing the model into the application.

- The resulting model was able to successfully predict 50 days in the future with a Multivariate RNN LSTM model as well as a successful univariate model that predicted exactly the true historical value for the next day. The data analysis section is a good addition to DOSAN as some analysts may simply want to check their data and use it in another program/method to build a model.

5.2 The business justification for this research

While this thesis focuses on the technical aspects and implementation, it is beneficial to evaluate the business opportunities that are relevant as the farming industry in the United States is $1.109 trillion [108]. The priority target market focuses on farmers and data analysts working in agriculture. However, this target market can be modified to a target market of developers and data analysts of any industry.

According to Statista, there is an estimate of around 2 million farms in the United States. 90% of these farms are small family-run farms and 10% of the farms are businesses. All the larger businesses will have internet access, while only 75% of the small family farms would have access to the internet. Therefore, 77.5% of all farms in the US have access to the internet and could benefit from the use of this technology. In the US, over four years a combined loss of the four of the main crops due to wildlife damage was estimated to be $592.6 million [109]. This technology would enable farmers to predict when infestations would occur that are linked to crop damage. This enables farmers to prepare and limit the damage through the use of natural solutions and pesticides, therefore, reducing loss of

earnings. As an example, if there was an Aphid infestation then the farmers may want to bring in the likes of Lacewings and ladybugs as a biological solution or deal with the manner using chemicals such as pyrethrum [110]. There is plenty of expansion with this technology as well in any industry which can discover hidden trends. As an example, expansion, a scenario could be in the medical industry where analysts are looking for a correlation between heart diseases and people. Analysts are perhaps trying to look at the industry-standard connections, but DOSAN can find the hidden trend.

Furthermore, a SWOT analysis, shown in Table 18, is useful for summarising the factors of CROM, RAW & DOSAN in terms of strengths, weaknesses, opportunities, and threats.

STRENGTHS	WEAKNESS
• Nothing directly similar in the market. • Highly accurate prediction forecasting. • The mobile application requires little processing power due to cloud support. • Reduce use of pesticides resulting in higher proportions of the crop being sold. • Reduce the time and effort needed for data analysts.	• Never been implemented in the market before.
OPPORTUNITIES	**THREATS**
• A solution to a worldwide problem in pest infestations. • Extending the use of not just agriculture allows for never found trends to come to light.	• Established companies creating equivalent system products.

Table 18 - SWOT Analysis.

Chapter 6 Bibliography

[1] Prabhu, "Understanding Hyperparameters and its Optimisation techniques," towardsdatascience, 03 July 2018. [Online]. Available: https://towardsdatascience.com/understanding-hyperparameters-and-its-optimisation-techniques-f0debba07568. [Accessed 01 04 2021].

[2] G. Thomas, "What is Flutter and Why You Should Learn it in 2020," freeCodeCamp, 12 12 2019. [Online]. Available: https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/. [Accessed 22 04 2021].

[3] D. Android, "Meet Android Studio," Developers Android, 18 05 2021. [Online]. Available: https://developer.android.com/studio/intro. [Accessed 20 05 2021].

[4] D. Stevenson, "What is Firebase? The complete story, abridged.," Medium, 24 09 2018. [Online]. Available: https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0. [Accessed 20 05 2021].

[5] G. M. Platform, "Maps SDK for Android overview," Developers Google, 18 05 2021. [Online]. Available: https://developers.google.com/maps/documentation/android-sdk/overview. [Accessed 21 05 2021].

[6] AccuWeather, "Current Conditions," AccuWeather APIs, n.d. [Online]. Available: https://developer.accuweather.com/accuweather-current-conditions-api/apis/get/currentconditions/v1/%7BlocationKey%7D. [Accessed 21 05 2021].

[7] A. Z. Mustafeez, "What is Visual Studio Code?," Educative IO, n.d. [Online]. Available: https://www.educative.io/edpresso/what-is-visual-studio-code. [Accessed 21 05 2021].

[8] M. Stojiljković, "The Pandas DataFrame: Make Working With Data Delightful," Real Python, 15 05 2021. [Online]. Available: https://realpython.com/pandas-dataframe/. [Accessed 21 05 2021].

[9] Safequard, "The Impact of Pests on the Agriculture Industry," Safeguard, 07 01 2020. [Online]. Available: https://www.safeguardpestcontrol.co.uk/impact-pests-agriculture-industry/. [Accessed 18 04 2021].

[10] P. J. R. A. J. M. R. T. T. S. J. I. A. J. P. F. M. C. J. M. Carlos Martínez-Núñez, "Direct and indirect effects of agricultural practices, landscape complexity and climate on insectivorous birds, pest abundance and damage in olive groves," *Agriculture, Ecosystems & Environment,* vol. 304, 2020.

[11] F. A. A. O. o. t. U. Nations, “The future of food and agriculture- Trends and Challenges,” Food and Agriculutre Organization of the United Nations (FAO), Rome, 2017.

[12] M. S. P. D. L.-T. P. K. K. W. G. A. Boursianis, “ Internet of Things (IoT) and Agricultural Unmanned Aerial Vehicles (UAVs) in smart farming: A comprehensive review,” ScienceDirect, Greece, 2020.

[13] M. Intel, “Farmers’ Growing Reliance on Technology Highlights Need for Robust Digital Toolbox,” Farm Burea, 26 08 2019. [Online]. Available: https://www.fb.org/market-intel/farmers-growing-reliance-on-technology-highlights-need-for-robust-digital-t. [Accessed 17 05 2021].

[14] A. M. Q. a. P. Cooper, “GPS-based Mobile Cross-platform Cargo Tracking System with Web-based Application,” *2020 8th International Symposium on Digital Forensics and Security (ISDFS),* vol. 1, no. 8, pp. 1-7, 2020.

[15] C. Software, “What is Flutter? Here is everything you should know,” Medium, 26 08 2019. [Online]. Available: https://medium.com/@concisesoftware/what-is-flutter-here-is-everything-you-should-know-faed3836253f. [Accessed 29 03 2021].

[16] C. Y. Y. L. S. T. a. S. H. W. Li, “JustIoT Internet of Things based on the Firebase real-time database,” *2018 IEEE International Conference on Smart Manufacturing, Industrial & Logistics Engineering (SMILE),* pp. 43-47, 2018.

[17] Yida, “Introduction to the Arduino – What is Arduino?,” Seeedstudio, 11 12 2019. [Online]. Available: https://www.seeedstudio.com/blog/2019/12/04/introduction-to-the-arduino-what-is-arduino/. [Accessed 29 03 2021].

[18] A. TEAM, “Welcome Raspberry Pi to the world of microcontrollers,” Blog.arduino, 20 01 2021. [Online]. Available: https://blog.arduino.cc/2021/01/20/welcome-raspberry-pi-to-the-world-of-microcontrollers/. [Accessed 29 03 2021].

[19] B. Jones, “Puppy Linux Review and its Status Quo in the Linux Community,” Foss Linux, 26 01 2021. [Online]. Available: https://www.fosslinux.com/43834/puppy-linux-review-and-its-status-quo-in-the-linux-community.htm. [Accessed 29 03 2021].

[20] Altexsoft, “The Good and the Bad of Ionic Mobile Development,” altexsoft, 21 05 2019. [Online]. Available: https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-ionic-mobile-development/. [Accessed 29 03 2021].

[21] J. a. C. D. Dahmen, “SynSys: A Synthetic Data Generation System for Healthcare Applications,” *Sensors,* vol. 19, p. 1181, 2019.

[22] J. L. a. J.-Y. L. a. L. Liao, “A new algorithm to train hidden Markov models for biological sequences with partial labels,” *BMC Bioinformatics,* vol. 22, 2021.

[23] K. Wiggers, “Generative adversarial networks: What GANs are and how they’ve evolved,” Venture Beat, 26 12 2019. [Online]. Available: https://venturebeat.com/2019/12/26/gan-generative-adversarial-network-explainer-ai-machine-learning/. [Accessed 08 04 2021].

[24] R. D. a. I. G. A. P. Y. K. B. Saloni Dash, “Synthetic Event Time Series Health Data Generation,” BITS Pilani, IIT Gandhinagar, UPSud/INRIA, Rensselaer Polytechnic Institute, Goa, Gandhinagar, Paris-Saclay, Troy, 2019.

[25] A. a. D. S. a. D. R. a. G. I. a. P. A. a. B. K. P. Yale, “Assessing privacy and quality of synthetic health data,” Association for Computing Machinery, New York, 2019.

[26] M. Holzapfel, “How is Tamr Different from MDM and ETL Tools,” Tamr, 28 06 2019. [Online]. Available: https://www.tamr.com/blog/the-difference-between-tamr-and-mdm-or-etl-tools/. [Accessed 12 04 2021].

[27] M. I. I. Stonebraker, “Data Integration: The Current Status and the Way Forward.,” *IEEE Data Eng. Bull.,* vol. 41, pp. 3-9, 2018.

[28] Guru99, “What is Informatica? Complete Introduction,” Guru99, 01 01 2020. [Online]. Available: https://www.guru99.com/introduction-informatica.html. [Accessed 22 04 2021].

[29] Vinod, “What is IBM?,” Techmonitor, 17 02 2017. [Online]. Available: https://techmonitor.ai/what-is/what-is-ibm-4950406. [Accessed 22 04 2021].

[30] Merk, “Merk,” Merk, Na Na 2021. [Online]. Available: https://www.merck.com/. [Accessed 22 04 2021].

[31] CSAIL, “Aurum is a data discovery system that works at large scale, helping people find relevant data.,” CSAIL, 16 08 2017. [Online]. Available: https://www.csail.mit.edu/research/aurum-large-scale-data-discovery. [Accessed 22 04 2021].

[32] G. H. S. E. W. Yuji Roh, “A Survery on Data Collection for Machine Learning,” *IEEE Transactions on Knowledge and Data Engineering,* vol. 33, no. 4, pp. 1328-1347, 2019.

[33] C. Verleyen, “What is Google Cloud Data Fusion?,” Medium, 24 04 2019. [Online]. Available:

https://medium.com/fourcast-premier-google-cloud-partner/google-data-fusion-63a45be48aa8. [Accessed 17 05 2021].

[34] H. Kaur, “What is Google Dataset Search and How to Use It?,” GeeksforGeeks, 02 05 2020. [Online]. Available: https://www.geeksforgeeks.org/what-is-google-dataset-search-and-how-to-use-it/. [Accessed 17 05 2021].

[35] Guru99, “Handling Dynamic Web Tables Using Selenium WebDriver,” Guru99, 07 05 2021. [Online]. Available: https://www.guru99.com/handling-dynamic-selenium-webdriver.html. [Accessed 17 05 2021].

[36] B. MADHUKAR, “The Continuous Bag Of Words (CBOW) Model in NLP – Hands-On Implementation With Codes,” Analyticsindiamag, 10 09 2020. [Online]. Available: https://analyticsindiamag.com/the-continuous-bag-of-words-cbow-model-in-nlp-hands-on-implementation-with-codes/. [Accessed 17 05 2021].

[37] H. a. W. J. Park, “CrowdFill: Collecting structured data from the crowd,” *Proceedings of the ACM SIGMOD International Conference on Management of Data,* pp. 57-90, 04 2014.

[38] V. a. M. P. a. Q. D. Crescenzi, “ALFRED: crowd assisted data extraction,” in *WWW 2013 Companion - Proceedings of the 22nd International Conference on World Wide Web*, Brazil, 2013.

[39] M. I. a. J. W. a. K. G. a. L. Zettlemoyer, “Adversarial Example Generation with Syntactically Controlled Paraphrase Networks,” *ArXiv,* vol. abs/1804.06059, pp. 1875-1885, 2018.

[40] J. Tanenbaum, “A First Look at HoloClean,” Medium, 15 07 2020. [Online]. Available: https://medium.com/@jacob.tanenbaum/a-first-look-at-holoclean-205ca7c71369. [Accessed 17 05 2021].

[41] L. C. M. A. S. R. S. a. T. I. R. A. Rashid, “Machine Learning for Smart Energy Monitoring of Home Appliances Using IoT,” *2019 Eleventh International Conference on Ubiquitous and Future Networks (ICUFN),* pp. 66-71, 2019.

[42] D. Cash, “THE RISE OF THE HOME ENERGY MANAGEMENT SYSTEM,” Installer Online, 21 12 2020. [Online]. Available: https://www.installeronline.co.uk/the-rise-of-the-home-energy-management-system/. [Accessed 25 04 2021].

[43] D. M. .Garbade, “What is Google Colab,” Education Ecosystem, 15 01 2021. [Online]. Available: https://blog.education-ecosystem.com/what-is-google-colab/. [Accessed 23 04 2021].

[44] S. Ray, “What is Matplotlib ? Basic Operations on MatplotLib,” Tutorialslink, 12 05 2020. [Online]. Available: https://tutorialslink.com/Articles/What-is-Matplotlib-Basic-Operations-on-MatplotLib/1404. [Accessed 24 04 2021].

[45] R. Kumar, “Matplotlib.animation.FuncAnimation class in Python,” GeeksforGeeks, 21 04 2020. [Online]. Available: https://www.geeksforgeeks.org/matplotlib-animation-funcanimation-class-in-python/. [Accessed 24 04 2021].

[46] Y.-H. a. C. C.-Y. Chang, “Classification of Breast Cancer Malignancy Using Machine Learning Mechanisms in TensorFlow and Keras,” in *Future Trends in Biomedical and Health Informatics and Cybersecurity in Medical Devices*, Cham, Springer International Publishing, 2020, pp. 42-49.

[47] S. Pal, “Scikit-learn Tutorial: Machine Learning in Python,” Dataquest, 15 11 2018. [Online]. Available: https://www.dataquest.io/blog/sci-kit-learn-tutorial/. [Accessed 25 04 2021].

[48] M. ,. P. a. Q. D. Chunqiao, “An Artificial Neural Network Approach to Student Study Failure Risk Early Warning Prediction Based on TensorFlow,” in *Advanced Hybrid Information Processing*, Cham, Springer International Publishing, 2018, pp. 326-333.

[49] Guru99, “Back Propagation Neural Network: What is Backpropagation Algorithm in Machine Learning?,” Guru99, 01 01 2020. [Online]. Available: https://www.guru99.com/backpropogation-neural-network.html. [Accessed 26 04 2021].

[50] D. Brownlee, “A Gentle Introduction to the Rectified Linear Unit (ReLU),” Machine Learning Mastery, 09 01 2019. [Online]. Available: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/. [Accessed 26 04 2021].

[51] S. Ray, “2019,” *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon),* pp. 35-39, 2019.

[52] M. B. a. H. M. J. a. N. S. R. H. Shams, “A Time Series Analysis of Trends With Twitter Hashtags Using LSTM,” in *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur,India, ICCCNT, 2020, pp. 1-6.

[53] T. B. a. K. V. a. N. S. a. S. S. a. R. S. a. S. Ghosh, “Earthquake trend prediction using long short-term memory RNN,” *International Journal of Electrical and Computer Engineering,* vol. 9, pp. 1304-1312, 2019.

[54] L. N. Smith, “A disciplined approach to neural network hyper-parameters: Part 1 - learning rate, batch size, momentum, and weight decay,” *ArXiv,* vol. abs/1803.09820, 2018.

[55] E. C. a. C. P. a. Z. P. a. N. P. a. S. R. a. M. A. Zinkevich, “TensorFlow Data Validation: Data Analysis and Validation in Continuous ML Pipelines,” in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, New York, 2020.

[56] P. K. DEVISSCHERE, “TensorFlow Extended (TFX): the components and their functionalities,” Adaltas, 01 03 2021. [Online]. Available: https://www.adaltas.com/en/2021/03/05/tfx-overview/. [Accessed 27 04 2021].

[57] beam.apache, “Apache Beam Overview,” Beam Apache, 24 02 2021. [Online]. Available: https://beam.apache.org/get-started/beam-overview/. [Accessed 28 04 2021].

[58] H. a. M. A. a. V. J. Weerts, “Importance of Tuning Hyperparameters of Machine Learning Algorithms,” Eindhoven University of Technology, Columbia University, Netherlands & New Yok, 2020.

[59] Simplilearn, “What is Azure and How Does It Work?,” Simplilearn, 12 02 2021. [Online]. Available: https://www.simplilearn.com/tutorials/azure-tutorial/what-is-azure. [Accessed 17 05 2021].

[60] Viscircle, “Beginner`s guide: What is OpenML?,” Visircle, 02 04 2020. [Online]. Available: https://viscircle.de/beginners-guide-what-is-openml/?lang=en. [Accessed 17 05 2021].

[61] H. a. L. S.-H. a. P. M. Yoon, “TensorFlow with user friendly Graphical Framework for object detection API,” University of Tasmania, Hobart, 2020.

[62] W. Y. S. P. F. L. Zewen Li, “A survery of Convolution Nerual Networks: Analysis, Applications and Prospects,” IEEE, Nanjing,China, 2020.

[63] T. H. R. K. Rafi, “Time Series Analysis - A Comparative Analysis Between ANN & RNN,” Daffodil International University, Bangladesh, India, 2020.

[64] O. -. O. D. Science, “Properly Setting the Random Seed in ML Experiments. Not as Simple as You Might Imagine,” Medium, 08 05 2019. [Online]. Available: https://medium.com/@ODSC/properly-setting-the-random-seed-in-ml-experiments-not-as-simple-as-you-might-imagine-219969c84752. [Accessed 22 04 2020].

[65] J. Brownlee, “Difference Between a Batch and an Epoch in a Neural Network,” Machine Learning Mastery, 20 07 2018. [Online]. Available: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/. [Accessed 22 04 2021].

[66] P. Gupta, “Cross-Validation in Machine Learning,” TowardsDataScience, 05 06 2017. [Online]. Available: https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f. [Accessed 22 04 2021].

[67] J. Brownlee, “Understand the Impact of Learning Rate on Neural Network Performance,” Machine Learning Mastery, 25 01 2019. [Online]. Available: https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/. [Accessed 22 04 2021].

[68] baeldung, “Epoch in Neural Networks,” Baeldung, 27 02 2021. [Online]. Available: https://www.baeldung.com/cs/epoch-neural-networks. [Accessed 22 04 2021].

[69] P. Karkare, “A 7 Minute Introduction to LSTM,” Medium, 22 11 2019. [Online]. Available: https://medium.com/x8-the-ai-community/a-7-minute-introduction-to-lstm-5e1480e6f52a. [Accessed 22 04 2021].

[70] M. Grootendorst, “Validating your Machine Learning Model,” TowardsDataScience, 26 09 2019. [Online]. Available: https://towardsdatascience.com/validating-your-machine-learning-model-25b4c8643fb7. [Accessed 22 04 2021].

[71] Guru99, “Unsupervised Machine Learning: What is, Algorithms, Example,” Guru99, 01 01 2020. [Online]. Available: https://www.guru99.com/unsupervised-machine-learning.html. [Accessed 22 04 2021].

[72] Guru99, “What is a Functional Requirement? Specification, Types, EXAMPLES,” Guru99, 26 05 2021. [Online]. Available: https://www.guru99.com/functional-requirement-specification-example.html#:~:text=A%20Functional%20Requirement%20(FR)%20is,%2C%20its%20behavior%2C%20and%20outputs.. [Accessed 28 05 2021].

[73] S. A. Framework, “Nonfunctional Requirements,” Scaled Agile Framework, 01 02 2021. [Online]. Available: https://www.scaledagileframework.com/nonfunctional-requirements/#:~:text=Nonfunctional%20Requirements%20(NFRs)%20define%20system,system%20across%20the%20different%20backlogs.&text=They%20ensure%20the%20usability%20and%20effectiveness%20of%20the%20entire%2. [Accessed 28 05 2021].

[74] R. Payne, Beginning App Development with Flutter- Create Cross-Platform Mobile Apps, Dallas: Apress, 2019.

[75] D. V. Kamani, “Flutter vs React Native in 2020: We Help You Decide What’s Best,” Arkenea, 02 March 2020. [Online]. Available: https://arkenea.com/blog/flutter-vs-react-native/. [Accessed 26 05 2021].

[76] N. Fuad, “10 good reasons to learn Dart,” Medium, 15 May 2019. [Online]. Available: https://medium.com/hackernoon/10-good-reasons-why-you-should-learn-dart-4b257708a332. [Accessed 28 05 2021].

[77] T. Contributor, “Android Studio,” TechTarget, 01 October 2018. [Online]. Available: https://searchmobilecomputing.techtarget.com/definition/Android-Studio. [Accessed 28 05 2021].

[78] I. Gaba, “What is Gradle And Why Do We Use Gradle?,” Simplilearn, 03 05 2021. [Online]. Available: https://www.simplilearn.com/tutorials/gradle-tutorial/what-is-gradle. [Accessed 28 05 2021].

[79] D. Stevenson, “What is Firebase? The complete story, abridged.,” Medium, 24 09 2018. [Online]. Available: https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0. [Accessed 28 05 2021].

[80] Simplilearn, “What is Tensorflow: Deep Learning Libraries and Program Elements Explained,” Simplilearn, 24 05 2021. [Online]. Available: https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-tensorflow. [Accessed 28 05 2021].

[81] Guru99, “What is TensorFlow? How it Works? Introduction & Architecture,” Guru99, 06 04 2020. [Online]. Available: https://www.guru99.com/what-is-tensorflow.html. [Accessed 28 05 2021].

[82] NumPy, “What is NumPy,” NumPy, 31 01 2021. [Online]. Available: https://numpy.org/doc/stable/user/whatisnumpy.html. [Accessed 28 05 2021].

[83] ActiveState, “What Is Pandas In Python? Everything You Need To Know,” ActiveState, 06 03 2021. [Online]. Available: https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know/#:~:text=Pandas%20is%20an%20open%20source,support%20for%20multi%2Ddimensional%20arrays.. [Accessed 28 05 2021].

[84] A. BHANDARI, “A Beginner’s Guide to matplotlib for Data Visualization and Exploration in Python,” Analytics Vidhya, 28 02 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/02/beginner-guide-matplotlib-data-visualization-exploration-python/. [Accessed 28 05 2021].

[85] FutureLearn, “What is Python used for? 10 practical Python uses,” FutureLearn, 09 04 2021. [Online]. Available: https://www.futurelearn.com/info/blog/what-is-python-used-for. [Accessed 28 05 2021].

[86] Guru99, “PyQt5 Tutorial: Design GUI using PyQt in Python with Examples,” Guru99, 28 04 2021. [Online]. Available: https://www.guru99.com/pyqt-tutorial.html. [Accessed 28 05 2021].

[87] CAST, “Risk Management in Software Development and Software Engineering Projects,” CAST - Software Intelligence for Digital Leaders, n.d. [Online]. Available: https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects. [Accessed 28 05 2021].

[88] Guru99, “What is Agile Testing? Methodology, Process & Life Cycle,” Guru99, 23 05 2021. [Online]. Available: https://www.guru99.com/agile-testing-a-beginner-s-guide.html. [Accessed 09 06 2021].

[89] Guru99, “What is Waterfall Model in SDLC? Advantages & Disadvantages,” Guru99, 28 05 2021. [Online]. Available: https://www.guru99.com/what-is-sdlc-or-waterfall-model.html. [Accessed 09 06 2021].

[90] GeeksforGeeks, “Difference between V-model and Waterfall model,” GeeksforGeeks, 21 05 2020. [Online]. Available: https://www.geeksforgeeks.org/difference-between-v-model-and-waterfall-model/. [Accessed 09 06 2021].

[91] Guru99, “V-Model in Software Testing,” Guru99, 03 05 2021. [Online]. Available: https://www.guru99.com/v-model-software-testing.html. [Accessed 09 06 2021].

[92] F. Turck, “The V-Model in Software Testing,” Froglogic, 16 06 2020. [Online]. Available: https://www.froglogic.com/blog/tip-of-the-week/the-v-model-in-software-testing/. [Accessed 09 06 2021].

[93] SoftwareTestingHelp, “What Is System Testing – A Ultimate Beginner’s Guide,” SoftwareTestingHelp, 30 05 2021. [Online]. Available: https://www.softwaretestinghelp.com/system-testing/. [Accessed 10 06 2021].

[94] SoftwareTestingHelp, “Performance Testing Vs Load Testing Vs Stress Testing (Difference),” SoftwareTestingHelp, 30 05 2021. [Online]. Available: https://www.softwaretestinghelp.com/what-is-performance-testing-load-testing-stress-testing/#1_Performance_Testing. [Accessed 12 06 2021].

[95] R. D. Melo, “Flutter Forms: Improving UI/UX with SingleChildScrollView,” Medium, 15 01 2019. [Online]. Available:

https://medium.com/@rubensdemelo/flutter-forms-improving-ui-ux-with-singlechildscrollview-7b91aa981475. [Accessed 12 06 2021].

[96] N. SCHÄFERHOFF, “Bootstrap Tutorial How to Setup And Use Bootsrap (Step-by-Step),” WebsiteSetup, 08 12 2020. [Online]. Available: https://websitesetup.org/bootstrap-tutorial-for-beginners/. [Accessed 12 06 2021].

[97] C. Hope, “HTML,” ComputerHope, 02 01 2021. [Online]. Available: https://www.computerhope.com/jargon/h/html.htm. [Accessed 12 06 2021].

[98] R. a. L. Y. a. G. J. a. K. P. a. S. M. a. E. N. Caruana, “Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-Day Readmission,” Association for Computing Machinery, New York, 2015.

[99] A. Brown, “Why Diversity is Essential for Quality Data to Train AI,” Techopedia, 08 02 2021. [Online]. Available: https://www.techopedia.com/why-diversity-is-essential-for-quality-data-to-train-ai/2/34209. [Accessed 14 06 2021].

[100] T. M. a. X. J. a. Z. Y. a. W. Pedrycz, “A survey on machine learning for data fusion,” *Information Fusion,* vol. 57, pp. 115-129, 2020.

[101] M. Chaudhary, “Why is Augmented Dickey–Fuller test (ADF Test) so important in Time Series Analysis,” medium, 09 04 2020. [Online]. Available: https://medium.com/@cmukesh8688/why-is-augmented-dickey-fuller-test-adf-test-so-important-in-time-series-analysis-6fc97c6be2f0. [Accessed 14 06 2021].

[102] R. A. G. Hyndman, Forecasting: principles and practice, 2nd edition, Melbourne, Australia: OTexts, 2018.

[103] D. Chan, “Why You Need Data Transformation in Machine Learning,” Datanami, 08 11 2019. [Online]. Available: https://www.datanami.com/2019/11/08/why-you-need-data-transformation-in-machine-learning/. [Accessed 14 06 2021].

[104] TensorFlow, “tf.keras.utils.normalize,” TensorFlow, 14 05 2021. [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/keras/utils/normalize. [Accessed 14 06 2021].

[105] TensorFlow, “Overfit and underfit,” TensorFlow, 12 03 2021. [Online]. Available: https://www.tensorflow.org/tutorials/keras/overfit_and_underfit. [Accessed 14 06 2021].

[106] J. Bronwlee, “How to use Learning Curves to Diagnose Machine Learning Model Performance,” Machine Learning Mastery, 27 02 2019. [Online]. Available: https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/. [Accessed 14 06 2021].

[107] R. Khandelwal, “A Basic Introduction to TensorFlow Lite,” TowardsDataScience, 01 06 2020. [Online]. Available: https://towardsdatascience.com/a-basic-introduction-to-tensorflow-lite-59e480c57292. [Accessed 08 06 2021].

[108] E. R. Service, “What is agriculture's share of the overall U.S. economy?,” U.S. Department Of Agriculture, 25 10 2020. [Online]. Available: https://www.ers.usda.gov/faqs/#:~:text=insurance%20premium%20subsidies.-,What%20is%20agriculture's%20share%20of%20the%20overall%20U.S.%20economy%3F,about%200.6%20percent%20of%20GDP.. [Accessed 16 05 2021].

[109] S. A. S. A. M. A. Sophie C McKee, “Estimation of wildlife damage from federal,” Society Of Chemical Industry, Fort Collins, 2020.

[110] L. Wyss, “How To Get Rid Of Aphids,” Insteading, 28 01 2021. [Online]. Available: https://insteading.com/blog/how-to-get-rid-of-aphids/. [Accessed 22 05 2021].

[111] T. d. A. C. Yambi, “ASSESSMENT AND EVALUATION IN EDUCATION,” ResearchGate, 2018.

[112] kahootz, “The importance of stakeholders in project management success,” kahootz, 12 02 2020. [Online]. Available: https://www.kahootz.com/why-stakeholder-management-is-an-important-part-of-project-management/. [Accessed 20 05 2021].

[113] Statista, “Number of farms in the U.S. 2000-2020,” Statista, 01 02 2021. [Online]. Available: https://www.statista.com/statistics/196103/number-of-farms-in-the-us-since-2000/#:~:text=In%202020%2C%20there%20were%20just,farms%20in%20the%20United%20States.&text=The%20average%20size%20of%20farms,been%20since%20the%20year%202000.. [Accessed 16 05 2021].

[114] E. R. Service, “Most farms are small, but most production is on large farms,” U.S. Department Of Agriculture, 02 12 2020. [Online]. Available: https://www.ers.usda.gov/data-products/chart-gallery/gallery/chart-detail/?chartId=58288. [Accessed 17 05 2021].

[115] E. S. a. M. I. System, “Farm Computer Usage and Ownership,” U.S. Department Of Agriculture, 16 8 2019. [Online]. Available: https://usda.library.cornell.edu/concern/publications/h128nd689?locale=en#release-items. [Accessed 17 05 2021].

[116] Castsoftware.com, “Risk management in software development and software engineering project,” 2019. [Online]. Available: https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects. [Accessed 17 November 2019].

[117] ReQtest, “Requirements Analysis - Requirements Analysis,” 2019. [Online]. Available: https://reqtest.com/requirements-blog/requirements-analysis. [Accessed 17 November 2019].

[118] TechTerms, “Computer Ethics,” 2019. [Online]. Available: https://techterms.com/definition/computerethics. [Accessed 21 November 2019].

[119] X. T.M, “A survey on machine learning for data fusion,” ScienceDirect, China, Finland, Canada, 2020.

[120] G. H. a. S. E. W. Y. Roh, “A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective,” *IEEE Transactions on Knowledge and Data Engineering,* vol. 33, no. 4, pp. 1328-1347, 2019.

[121] M. H. S. R. H. N. M. B. Shams, “ A Time Series Analysis of Trends With Twitter Hashtags using LSTM,” Daffodil International University, Bangladesh , 2020.

[122] W. Y. S. P. F. L. Zewen Li, “A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects,” IEEE, Nanjing, 2020.

[123] T. H. R. K. Rafi, “Time Series Analysis- A Comparative Analysis Between ANN and RNN,” Daffodil International University, Bangladesh, 2020.

[124] M. P. H. Y. S.-H. L, “TensorFlow with user friendly Graphical Framework for object detection API,” University of Tasmania, Hobart, 2020.

[125] G. L. Team, “What is Numpy in Python | Python Numpy Tutorial,” Great Learning, 28 05 2021. [Online]. Available: https://www.mygreatlearning.com/blog/python-numpy-tutorial/. [Accessed 17 05 2021].

[126] R. Berezhnoi, “What is Bootstrap and How to Use it in Web Development?,” F5 Studio, 18 01 2019. [Online]. Available: https://f5-studio.com/articles/what-is-bootstrap-and-how-to-use-it-in-web-development/. [Accessed 29 03 2021].

[127] T. Coron, “What is Sass? Your guide to this top CSS preprocessor,” CreativeBloq, 10 01 2020. [Online]. Available: https://www.creativebloq.com/web-design/what-is-sass-111517618. [Accessed 29 03 2021].

[128] Developers.Google, “Introduction to Gulp,” Developer.Google, 01 05 2019. [Online]. Available: https://developers.google.com/web/ilt/pwa/introduction-to-gulp. [Accessed 29 03 2021].

[129] Guru99, “What is AngularJS? Architecture & Features,” Guru99, 01 01 2020. [Online]. Available: https://www.guru99.com/angularjs-introduction.html. [Accessed 29 03 2021].

[130] S. Ulili, “The Beginner’s Guide to Chart.js,” Stanley Ulili, 05 10 2019. [Online]. Available: https://www.stanleyulili.com/javascript/beginner-guide-to-chartjs/. [Accessed 29 03 2021].

[131] J. Nunns, “What is Visual Studio?,” Tech Monitor, 11 04 2017. [Online]. Available: https://techmonitor.ai/what-is/what-is-visual-studio-4959054. [Accessed 25 04 2021].