Open University of Cyprus

Faculty of Pure and Applied Sciences

Master's Degree *Cognitive Systems*

Master's Thesis

Decision Support in Cognitive Production Planning

Andreas Paradisiotis

Supervisor

Loizos Michael

May 2021

Open University of Cyprus

Faculty of Pure and Applied Sciences

Master's Degree *Cognitive Systems*

Master's Thesis

Decision Support in Cognitive Production Planning

Andreas Paradisiotis

Supervisor

Loizos Michael

This master's thesis has been submitted as part of the requirements to obtain a master's degree
in Cognitive Systems
by the Faculty of Pure and Applied Sciences
of the Open University of Cyprus.

May 2021

Blank page

Abstract

In process manufacturing industries and more specifically in the poultry industry which deals with live stocks there are non-deterministic factors that affect the growing performance of a flock which make the planning process of picking and adding new flocks more complicated for a human being without having supporting information. We propose a system where through human machine interaction the user will create enough mental models, arguments and reasons until to come to the final decision, to accomplish the task of planning the daily needs for processing.

Contents

Chapter 1
Decision Support

In the field of manufacturing and process automation researchers since 1970 have focused on the development of decision support IT based systems which provide support to final the decision maker [1]. Human skills and computer capabilities required in the decision support loop. Typically, there is a source of data that will be feed into a decision system in order to provide by its turn analytical reports, suggestions or solutions to the decision makers. Traditionally companies are using Enterprise Resource Planning (ERP) systems to collect data through their daily process, and Business Intelligence systems (BI) to load, transform, analyze and visualize data to a decision maker. There are commercial and non-commercial ERP and BI systems which have incorporate DSS systems in the core of their applications and provide an end to end totally integrated solution from data acquisition to data visualization in order to provide decision support. Also, there are DSS systems that have been developed as monolithic systems and they provide an input interface which can be integrated in any system which can provide a source of data that will be analyzed further by the DSS. Finally, they provide an output interface which is the output of support data like suggestions, predictions, solutions, numeric values, and data models. Interactive DSS systems give the advantage to the decision maker to understand better the input and output data and create in its mind enough mental models about DSS system behavior. In the next chapters we will discuss about the concept of machine coaching and how machine coaching can be integrated into a DSS system and how the decision maker can provide guidance and coaching to the DSS in order to improve its support data. Massive amount of transactional data from process execution systems like Industrial Programable Logic Controllers (PLC) can be acquired through the process automation in a structured or unstructured format but these data cannot make any sense to any human being without any further analysis. The large amount of data makes the process of analysis not feasible for a human data analyst. Therefore, computational resources and machine learning techniques needs to be incorporated in order to derive meaningful data and further the generation of a meaningful data model.

Definition

A DSS is defined as a computer system which deals with a problem where at some point becomes semi-structured or unstructured [2]. Based on the structure of the task the system constructs a strategy which supports the cognitive processes of the individual decision maker to reflect the implementation strategy [3]. There are different variations of DSS,

- Passive DSS
- Active DSS
- Cooperative DSS

A passive DSS it only supports the process of the decision making without generation of suggestions or solutions. On the contrary an active DSS can provide decision making support and also generate suggestions and solutions. The cooperative DSS allows the decision maker to refine the generated suggestions based on his recommendations and beliefs and the system validates his suggestions and continually improves and adapts and digest the decision maker decision policy.

Decision Making Process

Decision making process can be described as the process of identification of possible solutions for an ill-structured problem with the aim to identify the best fit solution and not so many alternatives. There are multiple steps for the decision-making process:

- Identification of the problem and the people who are involved in the decision process, definition of the problem and expression of initial and desired conditions.
- Identification of requirements, constraints, and goals
- "Intelligent phase": Identification of possible alternative solutions which meet the defined requirements to draw a decision statement.
- Model development & alternative generation: Model development required for analyzing alternative solutions and based on the defined criteria the alternative solutions is evaluated based on their effectiveness. The best model is the one which achieve the defined goals and criteria better than the others. On the alternative analysis the evaluation of alternatives against criteria and the selection of decision making tool like Pros and Cons Analysis or Multi Criteria Decision Analysis etc. , is necessary to be chosen, however the ongoing research and experimentation with different decision methods and evaluations among them will show which decision method is the more appropriate to the problem [4] [5].
- Validation: Aims to validate the decisions taken by the system and ensure that truly solves the defined problem.

DSS Architecture

The major components of a DSS architecture is the database management system (DBMS) which is responsible to hold external and internal data of the system and can be accessed by other modules of the system. Furthermore, in the architecture we have the model management system (MBMS) which is responsible to reuse or make mathematical calculations for model construction and simulation purposes. This module handles the analysis of complex data sources and provide to the user interface component meaningful data [6].

DSS Application Development

- A Model driven DSS uses algebraic decision analytic, optimizations and simulation models for decision support [7]. These models are not data hungry and not relay in huge amount of data after the analysis. The user has the option to manipulate the model parameters to change the support behavior of the model. In general, these models try simplifying the real world analyzed situation. The decision analysis is presented with the use of statistical tools s such as analytical hierarchy process, decision tree analysis [8], multi-criteria decision analysis [9], [10] , and probabilistic forecasting [11] [12]. These methods discover the best alternatives on a given situation. Today there are various models available for supply chain management, including production planning and scheduling [13] [14] [15], demand management [16] [17], and logistics planning [18] [19] which are using optimization techniques to find the optimal alternative. Several kinds for simulation are the Monte Carlo simulation, discrete simulation, agent-based and multi-agent simulation, system dynamics, and visual simulation [20] [21].

- Data driven DSS relies on structured data which can be manipulated and can handle time-series of real time or historical data [22]. The amount of data depends on the problem but most of the time is huge. Example of data driven DSS are Executive Information Systems (EIS), Business Intelligence Systems (BI) and Online Analytical Processing Systems (OLAP) [23]. The main goal of these systems is to increase the quality of the information and the success relies in the accuracy of the data and at which level the data are organized and well-structured [24].

- Document driven DSS extract information from unstructured sources like videos, images, sounds, hypertext etc. [25]. The main goal is to extract information from unstructured sources and represent into a structured format for better decision support [26].

- Communication driven DSS relies on a distributed network where multiple decision makers collaborate and share information into a group of people so called Group Decision Support Systems (GDSS) [27] [28]. These systems provide tools to a group of people to formulate the problem in a structured manner. Collaborative Decision Support Systems (CDSS) [29] [30] are interactive computer-based systems where a group of decision makers searching for solutions and alternatives for a formulated problem [31]

- Knowledge driven DSS generally have their roots from Intelligent Decision Support systems and broader from Artificial Intelligence (AI) [32] [33]. These systems are computer-based reasoning systems which integrate AI technologies, expert systems [34] and data mining. Generally, they divided into two categories. The first is the rule-based systems which are using contemporary AI technologies and can handle pretty well reasoning and argumentation. This type of systems used effectively and widely for scheduling in production systems [23]. Expert systems rely on strategies defined with use of rules in a declarative way which will lead to a solution to a problem. Human expert knowledge needed with a set of facts from a database context to solve problems [35]. The second category contains earlier AI technologies like neural networks [36] , deep neural networks, fuzzy logic and genetic algorithms [37]. These are closer methodologies to linear programming modelling with more probabilistic intention.

Machine Coaching

Machine coaching is the process of human and machine interaction with the use of machine learning techniques and focus on the interaction between human to machine, machine to human on the level of both party's reason and express their beliefs and thoughts not necessary in a cognitive level that a human can reason and create arguments but in a level that both can understand each other. Furthermore, machine coaching can be adopted by declarative symbolling rule-based reasoning systems but broader can be used in supervised and semi-supervised learning systems where the human needs to assign simple labels on training instances. These methodologies provide very little information without enough quality arguments for creating an acceptable hypothesis and seems to be limited cause the information provided by the human expert is obviously not expressed in form that can be further processed. Also, these systems can be treated as black boxes because they don't have explanation capabilities from there nature. They key point in machine coaching is that the human and machine agreed to use a loose communication language that can be understood by both and through argumentation offer understanding of each other reasoning behavior [38].

Chapter 2
Problem

Definition

A system that is going to be used by people in agriculture poultry industry for advising users in their daily tasks of planning adding new flocks and picking daily needs for consumption to serve the market demand. Assuming having an input from information that captures the market demands like smart fridges connected to The Things Network [39] capturing in an intelligent way the daily needs of each house and then summarizing to each producer market insights about its products and the total needs, then the producer will be able to gain more accuracy in its planning process. We propose a system that through interaction with a user in its daily planning activities will gain knowledge and abilities to predict future plans for picking and placing activities and through a meaningful presentation of facts using visualization techniques make the user create enough mental models to take the grasp behind the reasoning in the plan. Finally, when users create enough mental models about the plan then they can intervene and make modifications in the behavior of the model.

Problem Analysis

Market demand

The system extracts the actual market demand from smart-fridges connected to The Things Network and creates a database and stores seasonal data with multiple product consumption analytics. These data used for future predictions on the market demand and these predictions help to estimate additions of new flocks. Predicting the future demand based on historical data and running a picking process simulation for one month will gave more insights to the producer when to put new flocks and make a cost-efficient plan to avoid running out of stock or the opposite to run out of storage.

Plant real time and historical data

These are real time statistics about the growing flock and can be divided into different categories of analysis.

Performance of the growing flock

These are KPIs for growth performance of a flock. Two very important metrics here are the FCR (feed conversion ratio) coefficient variation, daily mortality rate, and the carcass yield of broiler chickens. We will discuss in the next sections what are these KPIs and how can be used effectively.

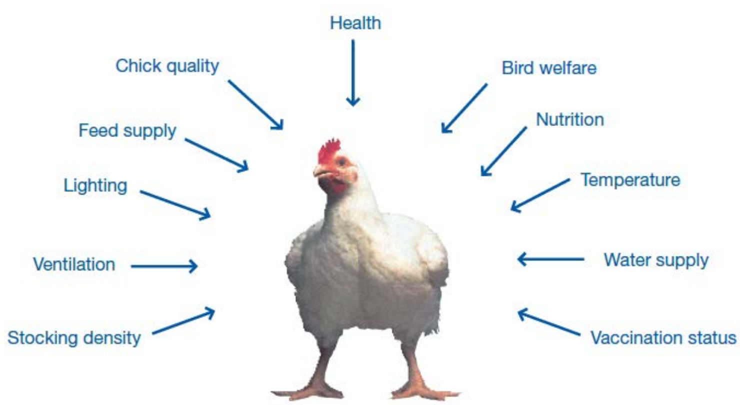


FIGURE 1 FACTORS AFFECTING BROILER GROWTH AND QUALITY [40]

Environmental conditions of the growing facilities

These are metrics like temperature, humidity, atmospheric pressure, ammonia levels and ventilation rates. These data can be used for monitoring the growing environment of the flock but also can be corelated with performance stats and make a prediction with input environmental data and target values performance KPI's. Here some papers with classification and regression techniques corelate environmental data with performance metrics to predict the future weight , FCR and yield. [41] [42] [43] [44]

Event	Records	Comment
Chick placement	Number of day-olds Flock of origin and flock age Date and time of arrival Chick quality Crop fill	Live weight, uniformity, number of dead on arrival Check crop fill percentage for age
Mortality	Daily Weekly Cumulative	Record by sex if possible Record culls and reason for culling separately Post-mortem records of excessive mortality Scoring of coccidial lesions will indicate level of coccidial challenge Record actual numbers and percentages Particular importance should be given to 7-day mortality
Medication	Date Amount Batch number	As per veterinary instruction
Vaccination	Date of vaccination Vaccine type Batch number	Any unexpected vaccine reaction should be recorded
Live weight	Weekly average live weight Weekly uniformity (CV%)	More frequent measurement is required when predicting processing weight.
Feed	Date of delivery Quantity Feed type Feed form Date of starting feed withdrawal prior to catching	Accurate measurement of feed consumed is essential to measure FCR and to determine cost effectiveness of broiler operation Check feed quality
Water	Daily consumption Water to feed ratio Water quality Level of chlorination	Plot daily consumption in graph form, preferably per house Sudden fluctuation in water consumption is an early indicator of problems Mineral and/or bacteriological especially where bore holes or open water reservoirs are used
Environment	Temperature: • Floor temperature as well as litter temperature -daily minimum -daily maximum -during brooding, 4 to 5 times per day -litter during brooding -external temperature (daily) • Relative Humidity (daily) Air quality Litter quality Last calibration of equipment and by who	Multiple locations should be monitored, especially in chick litter area Automatic systems should be cross-checked manually each day Ideally record dust, carbon dioxide (CO_2) ammonia (NH_3) or as a minimum observe levels of dust and NH_3
Depletion	Number of birds removed Time and date of removal	
Information from processing plant	Carcass quality Health inspection Carcass composition Type and % condemnations	
Cleaning out	Total bacterial counts	After disinfection, Salmonella, Staphylococcus or E. coli may be monitored if required
House inspection	Record time of daily checks Make note of any bird observations	Behavior and environmental conditions
Lighting program	Dark and light period Time on and time off	Intermittent or not
Visitors	Who Why Date and reason for visit Previous farm visits (place and date)	Should be completed for every visitor to ensure traceability

FIGURE 2 RECORDS REQUIRED IN BROILER PRODUCTION [7]

Historical Incidents data after slaughtering

After slaughtering process, the product is inspected by the quality assurance team by taking a random sample for each flock and identify in detail specific issues. Also, they measure the average yield for each flock after the end of slaughtering process. Then they update the incident database with the new findings. From that data we can create a model using classification and regression for each flock/plant and then we can have a model to make inferences.

Picking Plan Generator

This part of the system is critical for creating a future plan for picking chickens from plants and transport to the slaughterhouse for processing.

Placement Plan generator

When the system generates the picking plan there is an option to generate also the placement plan in parallel. Generation of the placement plan happens after an estimation of when a plant will run out of stock in relation with future market demand.

Explanation and Coaching View

The explanatory, coaching view provides explanations about the generated picking and placement plan. For example, the user can make questions why the system chooses to pick or place stock on a specific plant and why not to choose an alternative one indicated by the user. The system will display in the explanation view useful visual representation and metrics comparing the two plants and showing with numbers and visualizations why it comes to the certain decision.

Business process

The figure bellow shows the traditional process for broiler meat production from the hatcher to retail. For the purposes of our system, we care about the operations that take place into the broiler farm and processing plant. We will further analyze using business terms the processes that we care about for the purposes of our system.

FIGURE 3 PRODUCING QUALITY BROILER MEAT – THE TOTAL PROCESS [7]

Chicken Placement into the farm

Usually there are hatchers who are responsible for the process of hatching broiler breeds and also handle the transportation to the broiler farms. There are various brands of broilers breeds which are providing the genetics and product performance guidance to the hatcher and the grower. Growers and hatchers have close communication when planning their placements of new flocks. Growers making first a placement plan usually for a whole year and during the year they adjust their plans accordingly. Hatchers needs to know the yearly plan of their partner grower in order to estimate their resources. The grower needs to confirm two months before the need of newborn chicks for adding a new flock to the farm and the hatcher producing the new flock of newborns and transport them to the grower. During the process of hatching, hatcher grower stays in communication and discussing delays and other details that may affect their plans.

FIGURE 4 TYPICAL CONTROLLED-ENVIRONMENT CHICK DELIVERY VEHICLES [7]

When a plant growing cycle ends growers are keeping the plant empty for an amount of time and during these "rest" period the plant will be cleaned up from diseases and will be prepared for the next flock. Approximately the period is around 40 days but as long as you are keeping empty the plant then better.

Figure 5 Examples of good biosecurity procedures [7]

One important factor that must be taken into consideration before adding a new flock are the environmental conditions in the current season and dimensions and availability of space of the plant which affect the decision of the stock density and influence directly the product quality, uniformity and bird welfare and product quality. There are various instructions from industry experts, broiler breed vendors and law enforcement experts that need to be followed carefully. Climate conditions and economics are also important factor to take the density decision. For example, in the hot seasons the density needs to be lower in contrast with the cold season where the density is higher.

Below is an example from the EU Broiler Welfare directive (2007) [40]:

> *33 kg/m^2 (6.7 lb/ft^2) or*
>
> *39 kg/m^2 (8.0 lb/ft^2) if stricter standards are met or*
>
> *42 kg/m^2 (8.6 lb/ft^2) if exceptionally high welfare standards are met over a prolonged period of time.*

Next a new flock added to the plant and the new cycle begins.

FIGURE 6 EXAMPLE OF GOOD-QUALITY CHICKS, FIGURE 7 CHICK PLACEMENT [7]

Flock growth management

Stockmanship is the positive outcome of the effective attention showed by the stockman in charge to manage the broilers and the plant environment to achieve effective growth performance by eliminating potential risk and issues during the process and helps to the maximum the objective to carry out on the market a very high quality product. The stockman always needs to be aware and use all of its senses capturing the broilers condition by constantly paying a lot of attention on the flock behavior in detail and actively using empirical techniques to identify and solve discrepancies. The chicken welfare rules need to be followed by the stockman during the interaction with the birds and always needs to expose care, dedication, staying up to date with new techniques and processes told from industry experts and constantly learning and enriching its skills on the field.

The stockman is responsible to identify remove and record daily dead chickens, make a manual weekly sampling of chicken weight by taking a random sample from the farm with the use of an electronic scaling machine and record the weekly growth and calculate the flock uniformity. Additionally, if the plant contains a controller device, then the weighing information and stock uniformity can be captured in real time by the controller.

FIGURE 8 EXAMPLE OF BIRD SAMPLE POINTS FOR WEIGHING. THE RED CIRCLES SHOW WHERE A SAMPLE OF BIRDS SHOULD BE TAKEN [7]

FIGURE 9 AUTOMATIC WEIGHING [7]

FIGURE 10 MANUAL BULK WEIGHING OF CHICKS WITH AN ELECTRONIC SCALE [7]

Finally, is responsible for making the daily empirical checks and inspect the water and feed quality, investigate and recognize diseases on individual birds and always be proactive and find solutions to decrease the risk of disease. Furthermore, needs to be strict on the vaccination plan and use the correct pharmaceutical treatment. Always needs to stay in close contact with stockman supervisor and inform him about the overall picture of the flock and take further instructions to overcome issues.

Feeding is one of the most important factors that influence growing performance. There are various different plans for broiler nutrition with combination of different ingredients used in the growing process. The size and form of the feed differs in different bird ages and the producer of the feed needs to pay attention on the specific broiler breed nutrition specifications.

Age (Days)	Feed Form	Particle Sizes
0-10 days	Sieved crumble	1.5-3.0 mm diameter
	Mini-pellets	1.6-2.4 mm diameter 1.5-3.0 mm length
11-18 days	Mini-pellets	1.6-2.4 mm diameter 4.0-7.0 mm length
18 days to finish	Pellets	3.0-4.0 mm diameter 5.0-8.0 mm length

FIGURE 11 FEED FORM AND RECOMMENDED PARTICLE SIZE BY AGE IN BROILERS. [40]

The person responsible creating the nutrition plans need to be in closed contact with stockman and discuss about the performance of the flock. Sometimes the nutrition planner may change the nutrition plan to make the flock grow faster or slower by adjusting quantity of ingredients that affect energy, proteins and vitamins indicators. Also, if a disease identified on the flock the nutrition plan may change so that a special nutrition plan may alter the situation and speed up the recovery time. In some cases, during periods with high risk disease affection antibiotics may used on the nutrition plan in order to mitigate the transmission of the disease.

Finally, as been said before, the water and feed quality are the most important factors that regulate the growing performance and always the grower needs to evaluate feed and water quality before consumption.

Plants with latest technology are equipped with computer controllers to control the ventilation inside the farm through control over the speed of air and operation of the air-

Commented [AP1]:

cooling system for cold and hot air, control the light intensity on different hours and at various flock cycle stages, and finally control the feeding and watering system to achieve the daily target feed and water quantity needed to be consumed by the flock. Silos are equipped with load cells which are measuring the current quantity of feed on the silo. A water flow meter used to measure the consumption of water and pH meter measures the pH of the water.

Also, these controllers incorporate various sensors for measuring temperature, humidity, dust level, atmospheric pressure, ammonia level, water quality, weighing sensors to measure the average flock weight, coefficient variation, state of the ventilation system and sensors capturing the daily consumption of water and feed with indications on the quality. There are special alarms that alert the stockman for discrepancies and issues that may potentially arise.

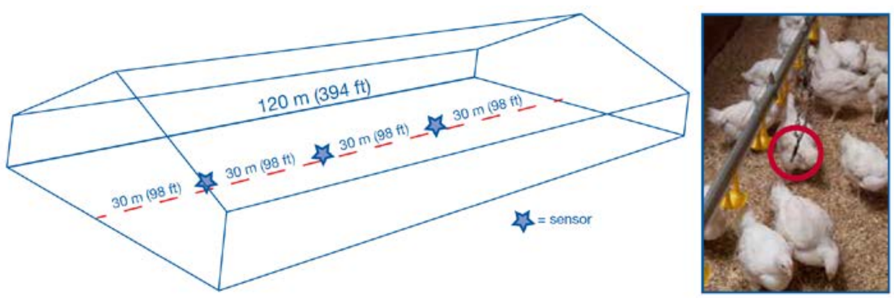


FIGURE 12 SENSORS LOCATIONS FOR TEMPERATURE AND HUMIDITY [45]

The system that we suggest is going to collect on a daily basis performance and environmental data from different plants that have installed a computer controller. The goal is to acquire these data on a central database and will be further analyzes and visualizes into a dashboard view. Further the system will have access to performance statistics that released by the broiler breed suppliers and the scoring module will continually modify the score for each plant based on weights given by the user on factors that affect the overall score. In the figure bellow we can see that a stockman uses its five senses to create mental models about the flock behavior. With the proposed system we added another "sense" which is access to meaningful analytics. The stockman manager using these analytics gain access to more insights about performance of each flock and influence him on planning stock transportation for slaughter process and addition of new flocks. Insights from performance perspective encourages the process of creating enough mental model that will lead the user recognize potential issues and flocks where need more attention because they don't meet the performance expectations.

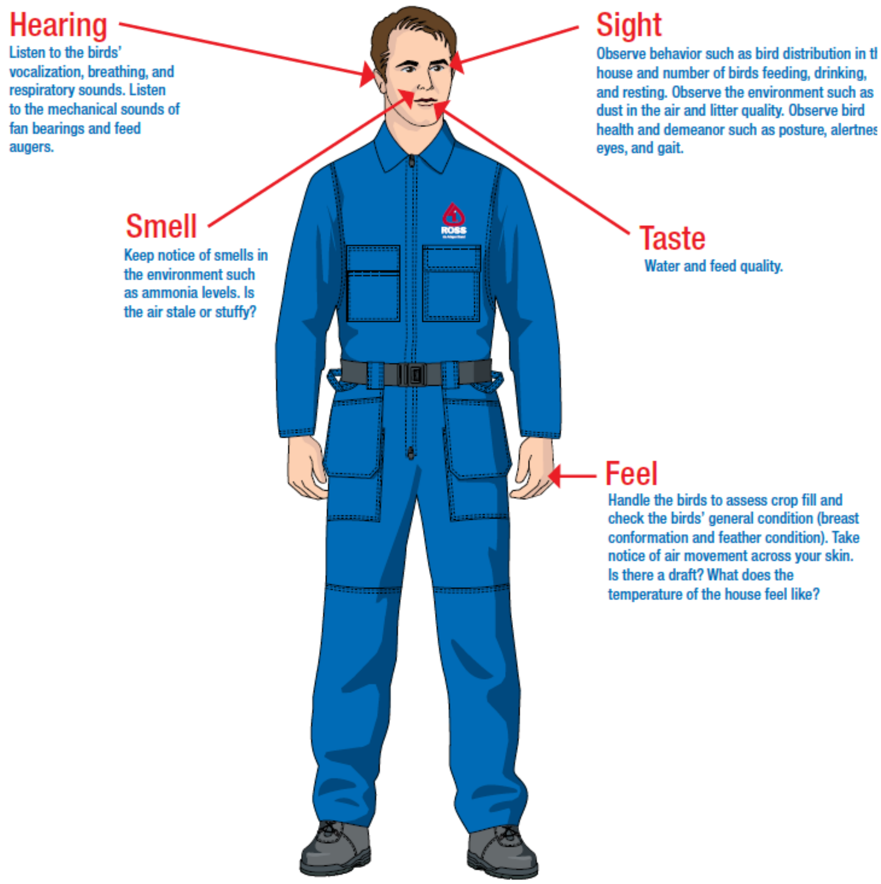


FIGURE 13 STOCKMANSHIP USING ALL THE SENSES TO MONITOR A FLOCK [40]

Choosing plants for catching chickens and transport to the slaughterhouse

The sales department specifies the target demand on a weekly and daily basis and indicate using constrains on quantity, weight, and class the load needed to be transported from the plants to the processing plant for slaughtering. The planner tries to find plants that can serve the specified constraints and tries to create a picking plan that will be cost efficient to transportation and meets the transportation constraints in terms of storage space and number of available truck and drivers. Factors like stock uniformity plays an important role on plant selection. If the uniformity spread is high, then you have more luck to find different weight scales that you need to catch but from the other hand these may indicate that the current flock may have some growing performance issues since the flock is not growing uniformly and needs more attention to mitigate potential issues. Also, historical incidents and performance metrics like the yield produced from each plant have to be taken into consideration to choose correct classes of product quality and meet the weight constraints.

FIGURE 14 EXAMPLE OF A VEHICLE SUITABLE FOR TRANSPORTING BROILERS TO THE PROCESSING PLANT [40]

Important metrics

There are a lot of quantified metrics that can be recorded during a breeding cycle. Here we want to mention which will be taken into consideration on the design step. We will split these metrics into two categories. The first category has to do with metrics that indicate performance and the second has to do with environmental metrics.

Performance metrics

FCR - Feed conversion ratio shows the efficiency of a bird to convert the feed consumed into live body weight.

$$FCR = \frac{Total\ Feed\ Consumed}{Total\ Live\ Weight}$$

The lower the FCR then the efficiency is higher. Is important metric and will be used from our system to predict the future FCR and estimate the days needed to achieve the live target weight.

Adjusted FCR - Adjusted feed conversion ratio is a formula which provides a good estimation of adjusted FCR.

$$Adjusted\ FCR = Actual\ FCR + \frac{Target\ Body\ Weight - Actual\ Body\ Weight}{Factor}$$

The factor here is constant value of 4500 kg. It has accuracy on target weight +/- 0.5 from the actual weight but outside the range the calculation is not approaching very well. Our system for short weight gain prediction we are going to use the adjusted FCR but for target weights that goes outside the range of accuracy we are going to use regression techniques.

There are various performance cards published by the breed suppliers and indicate daily performance statistics for specific breed. These cards can be used to make comparisons with the live flock and see if the flock is approaching the target weight and expected FCR.

PEF - Production Efficiency Factor

$$\frac{Livability\ X\ Live\ Weight\ in\ kg}{Age\ in\ Days\ X\ FCR}\ X\ 100$$

The higher value the better technical performance. Livability is the percentage of the initial stock minus the dead birds during the cycle. This metric is high biased by daily gain factor and is good to be used in situations where we need to compare for example two plants efficiency factor on the same age.

CV% - Coefficient of Variation Ratio is the variability of population in flock and is formulated by the standard deviation of the population projected as percentage of the mean. Variable flocks have higher CV% and uniform flocks have lower one.

$$\frac{Standard\ Deviation}{Average\ \ Body\ Weight}\ X\ 100$$

The figure bellow shows the number of birds needed to sample from a flock to get an accurate estimate. As the CV% increase then number of birds to be weighted increase also.

Uniformity of Flock+	Number of Birds to be Weighed++
Uniform (CV% = 8)	61
Moderately Uniform (CV% = 10)	96
Poorly Uniform (CV% = 12)	138

FIGURE 15 MINIMUM NUMBER OF BIRDS IN SAMPLE TO GIVE ACCURATE ESTIMATES OF LIVE WEIGHT ACCORDING TO FLOCK UNIFORMITY.

The accuracy of the CV factor is 95% correct if the estimate of the live weight is +/- 2% of the actual weight. The CV% factor is very important for our system and will be taken into consideration to estimate the uniformity of the flock on a target age.

Yield is a performance ratio acquired after the slaughtering process and indicating the meat or carcass extracted from the body of the whole chicken. For example, the weight of the chicken may be three kilos but after processing in the weight definitely will be lower, because of the removal liquids, bones, feathers and other parts of the body [46]. In appendix 1 the last figure shows the dry yield performance of Ross308. Yield depends on the performance and efficiency of the end-to-end process of growing, catching, and processing.

CY -> Carcass yield

CW -> Carcass weight

BWs -> Body weight before slaughter

$$CY(\%) = \frac{CW}{BWs}\ X\ 100$$

Mortality rate is a metric that indicates how many chicken's dead through the growing cycle including the processing stage of the whole population. It can be projected in different ways like 2% of the population or on average like 5 dead chickens per day. Is an important metric indicating in the early stages potential issues. If the indication is high that means that you need activate other policies to identify the cause root of the problem. Is good practice to separate mortality counting in the growing and processing plant and these may give a better picture and facilitates the process to identify the environment that causes these issues.

Various *incidents* are sampled after the slaughtering process to identify issue on quality of the product. During the inspection process all the incidents are recorded with a specific class and subclass of incident, date, number of the cycle and the name of the growing plant. These data can be further analyzed and identify the rate of incidents in each plant and after assign weights on each incident the overall score of each plant can be calculated.

For the purpose of our system the scoring system and in general the classes of incidents appear on each plant cycle will be fed in a learning algorithm to predict potential incidents in the future. The class of incident may indicate the quality of the stock, and this will help the system to make further assumptions for example where to find chickens for pickup with a specifics product quality class.

	ΜΟΛΥΝΣΕΙΣ		ΦΤΕΡΟΥΓΕΣ		
	ΔΕΡΜΑΤΙΚΗ ΣΤΟ ΣΤΗΘΟΣ	ΣΤΙΣ ΦΤΕΡΟΥΓΕΣ	ΚΟΚΚΙΝΕΣ	ΜΑΥΡΕΣ	ΣΠΑΣΜΕΝΕΣ
03/02/2020	50		30		20
06/01/2020	10	20		10	
18/02/2020			80	20	
31/01/2020					20
Grand Total	30	20	55	15	20

FIGURE 16 INCIDENT DASHBOARD SHOWING THE AVERAGE COUNT OF EACH CLASS AND SUB CLASS

Environmental metrics

We consider environmental metrics, quantified measures that can be extracted from the controlled environment like temperature, humidity, ammonia level and atmospheric pressure. For the purpose of our system, we want to corelate environmental metrics with performance metrics to identify relations between the two types and make predictions on performance of the flock by providing in the inference environmental metrics.

Initial Designs/Mock-Ups of System UI Components

We propose a web based centralized system where all the available sources of data will be analyzed and visualized to the end user. Through basic interaction with the system our goal is to make the user create a good overall picture of how several flocks are performing and through visualization of meaningful UI elements like Gantt charts, calendars, maps, performance tables and other visual elements that make the user get a quick indication about overall performance of the flocks. Additionally, we propose an interactive method where the user will give feedback to the system by easily adjusting the classification parameters and actively informing the system for new changes that may miss or not having enough resources to analyze and come to a conclusion. Also, we want to give to the user the ability to adjust rules using traditional symbolic rule-based approaches and these will help in the process of planning adding and collecting amounts of chickens from the plants. Using traditional constraints satisfaction techniques, variable manipulation, route cost estimation, scoring, decision making and in general techniques that can satisfy the process of planning using simple rules.

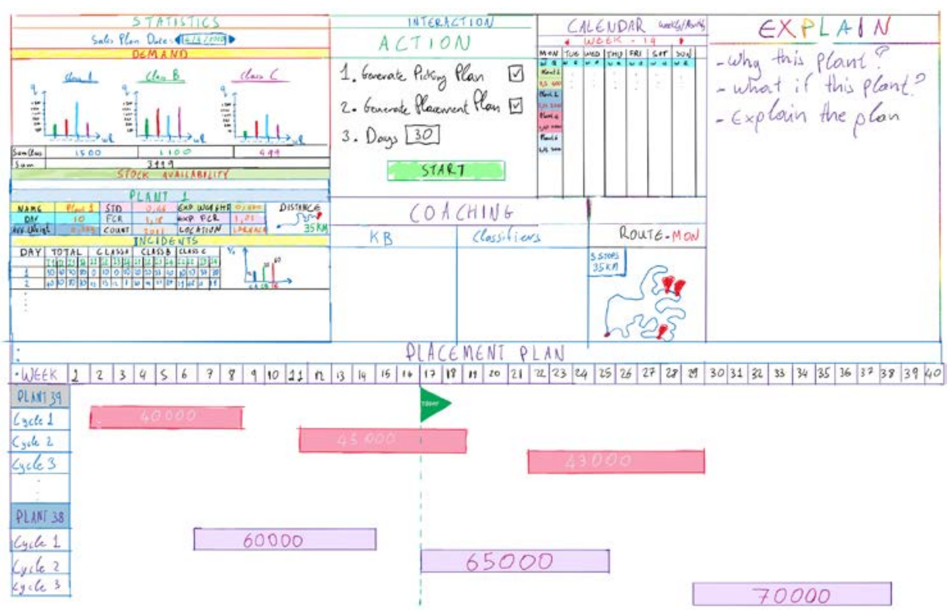


FIGURE 17 MAIN USER INTERFACE

The figure above shows the basic idea of the user interface. Next, we want to describe our main UI components and later the system architecture.

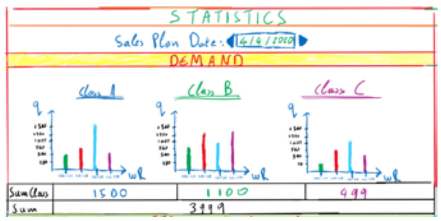


Figure 18 Sales demand data with quantity and weight constraints

Sales plan statistics

In the figure above the user can see the daily created sales plans created by the sales department according to the market constraints. Further the user can dig into more details on the exact quantities needed for each weight range and class. So here we can see that constrains likes quantity, weight range, and class are important measures in the process of identifying plants that can satisfy these constraints.

Stock availability and performance data

In the figure bellow we can see the panel where the user can navigate through active individual plants and get real time insights and summarized analytics about the performance and overall statistic of stock count, current day of growing, average FCR, coefficient variation to indicate uniformity of the plant and average weight of the flock of each plant. Bellow the stock availability and performance metric stats helps the user to monitor the daily sampled data from quality inspections after the slaughtering process. Later we will indicate how will corelate current and historical data to predict the quantities of classes of each flock, the weight and uniformity factor. When we say class we mean the quality class of the chicken which usually is spitted into three categories of A,B and C. The responsible person for categorizing the slaughtered chickens usually is doing an eye contact inspection to identify quality issues.

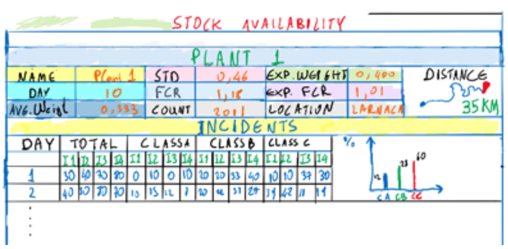


FIGURE 19 STOCK AVAILABILITY

Action View

In this view the user can specify which plans wants to generate and for how many days. It can choose to generate the picking plan or both picking and placing. Is good to mention here that if the user needs to generate the placement plan, then is mandatory for the system to generate also the picking plan in order to simulate when plants are going to run out of stock and based on those calculations the system will generate the future placement plan. When the user hit the start button the system will start generating those plans.

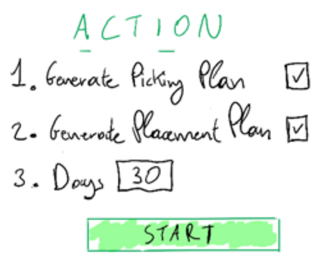


FIGURE 20 ACTION VIEW

Picking plan calendar view

After the generation of the picking plan by the system with the use of the predictive and rule-based algorithms, the plan needs to be visualized to the end-user in a meaningful style. For that purpose, the figure bellow shows a calendar view which can be presented in weekly and monthly fashion and present to the user the generated plan. In the figure we see the quantities per weight range needed to be taken from each plant to satisfy the sales demand.

FIGURE 21 CALENDAR VIEW

When the user clicks on a calendar day the routing view will display the routing plan for the corresponding day with an estimation of the distance and stops.

FIGURE 22 MAP OF THE ROUTES

Placement plan Gantt chart

When the user chooses to generate the picking plan the algorithm first will start running in simulation the picking plan to estimate when certain plants are going to run out of stock and further start calculating in which day and in which plant a new flock must be added with specific quantity. In the figure bellow we present a Gantt chart for that purpose and on the row level we indicate each plant with its cycled as sub items. Usually, a broiler plant can make 6 cycles per year including the death periods and each cycle have 45 -50 days approximate duration. On the column level we display the number of the week during the year and each placement is visualized as a horizontal bar with specific duration and quantity on the cycle level. This view will assist the user to get an overall picture about his historical and future placements.

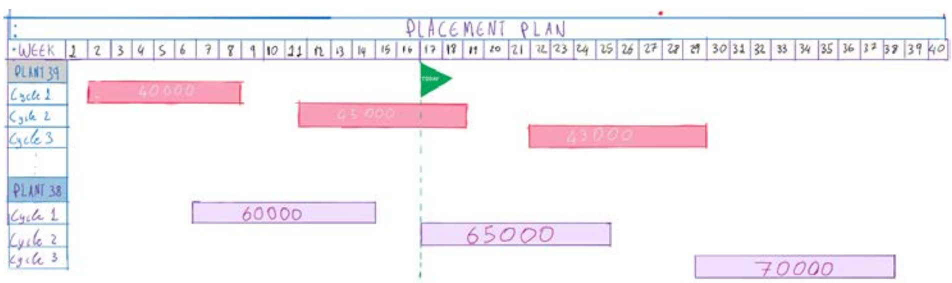


FIGURE 23 PLACEMENT VIEW

Interaction / Explanation View

In the figure bellow we present the coaching and explanation view. Basically, in these views the user will be able to interact with the system. The coaching view is spitted into two sections. The knowledge base and classifiers. In the knowledge base the user will be able to add custom rules that will be taken into consideration when the system is generating the plan. For example, one rule for the placement plan can be the death period needed for the plant to be empty after the end of a cycle. In the classifier section the user will be able to adjust classifier parameter and indirectly create or modify training instances. For example, in the case that we need to predict the future mortality rate of the plant using historical data the user will be able to adjust the training instances and indirectly inform the system that the future mortality rate will be increased or decreased.

In the explanation view the user will get explanation why the planning module take the decision to pick up or place birds in the plant. Additionally, the explanation view will provide explanations about the performance of the flock. For that purpose, we split these requirements into four questions.

1. *Why you choose this plant? (single plant explanation)*
2. *What if this plant? (comparison between two plants)*
3. *Explain the whole plan (summary explanation)*
4. *Indicate performance and environmental issues and warnings.*

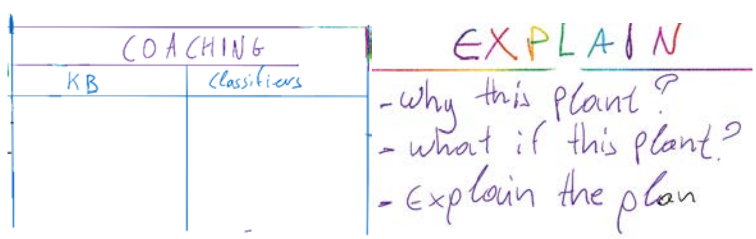


FIGURE 24 COACHING VIEW

FIGURE 25 EXPLANATION VIEW

Example scenarios for each question

For the first question *Why, you choose this plant?* the user will be able to right click on a predicted placement or picking plant from the Gantt chart or the calendar view and with a right click on the context menu will click on the particular question. Then in the explanation view will get the following answer.

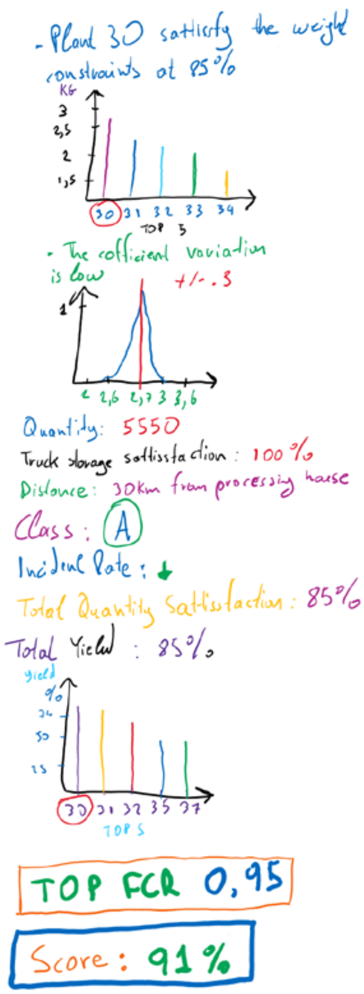


FIGURE 26

EXAMPLE OF ANSWER ON THE QUESTION WHY YOU CHOOSE THIS PLANT?

In the figure above we illustrate an example of an answer. First the assistant shows to the user a graph with average weights of all the flocks and indicates to the user that the selected plant satisfies the constraints better than other plants. Then the assistant shows to the user a graph that shows the uniformity of the flocks and indicates to the user that the coefficient variation of the flock is low. Then it shows to the user some important metrics that taken into consideration like the quantity needed, truck storage capacity satisfaction, the distance from the processing house, the class of product quality and incident rate which is derived from the incident data sampled after slaughtering. Finally, it shows graph with yield performance of each flock and a KPI on the FCR and total score. The total score is calculated by the system using environmental, and performance data. We will discuss later on how the total score is calculated.

The second question that a user can make to the system is “what if you choose this plant?”. It sounds like a comparison question and uses in situations where the user wants to compare the selected plant chosen by the system with a plant of his choice. The idea here is to visualize to the user graphs that make comparisons between the two plants and indicate important metrics why the system takes the decision to choose a particular plant over the other. The user again needs to make a right click on the suggested plant and on the context menu to choose the appropriate question. Then the system will ask the user which plant wants to compare.

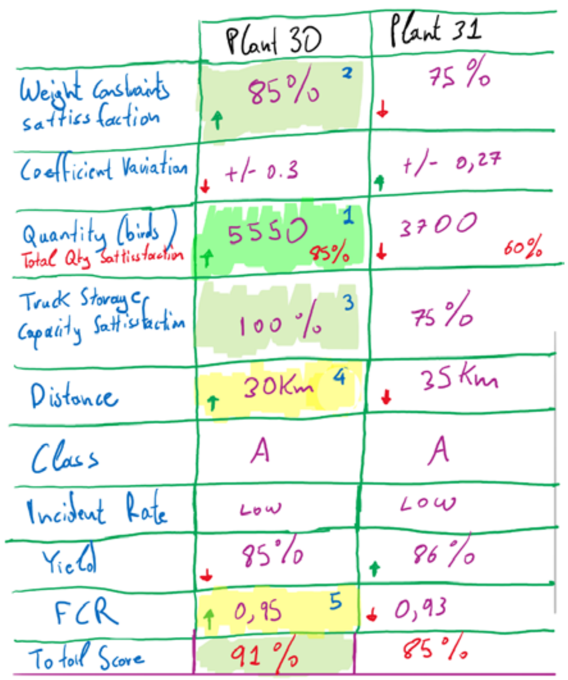


FIGURE 27

COMPARING PLANTS WITH DIFFERENT FACTORS THAT INFLUENCE THE FINAL DECISION

In the figure above we show an example of answer, where the system visualizes a comparison table with various factors that affect its decision and giving to the user explanatory facts behind that influence the decision. The green and re arrows indicate the winner of the corresponding factor and when a fact is highlighted it's an indication that the corresponding factor has influenced the overall decision. In the example above the quantity of birds it's the most influential factor and ranked as the number one factor. Then we see that the weight constraints are satisfied at 85% in plant 30 of the overall quantity needed. Next, we see that the storage of the truck will be consumed 100% and this factor seems to be important because the user may prefer to sending trucks to plants that can consume the full capacity of the storage.

The user may need to get insights about the current performance of running flocks. For that purpose, the system will visualize environmental and performance data and warns the user about potential or current issues in flock by looking for correlations between performance and environmental metrics which indicate potential issues. In the figure bellow we show a dashboard from a software which visualize environmental and performance data and with the arrows with red color we indicate a correlation between mortality rate and temperature which shows when the temperature is over specific range leads to higher mortality rate which make sense since the birds can leave under specific environmental conditions. Other factors can indicate higher mortality rates, like identification of a disease, stock density or the recommended levels of static pressure and CO2 are not met. Further feed and water quality can be correlated with mortality rate. We want to provide to the user visualizations of timeseries graphs and draw on them correlations, warnings and explanations and indicate instantly to the user the important spikes where need more attention.

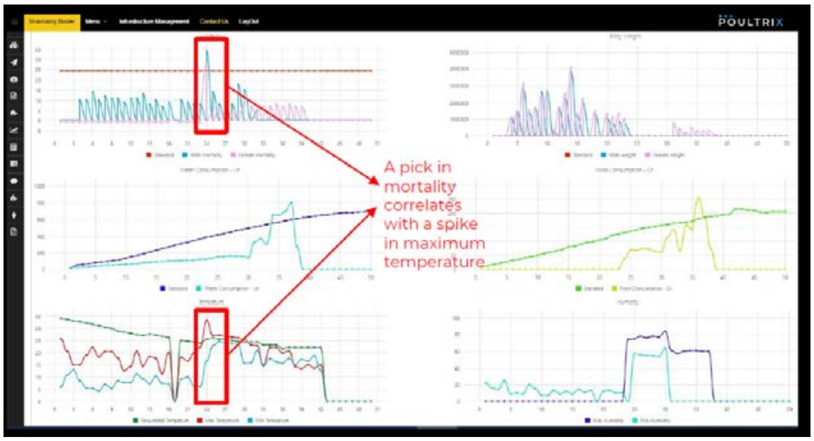


FIGURE 28 POULTRIX SOFTWARE PERFORMANCE AND ENVIRONMENTAL GRAPHS

Chapter 3
Solution

We propose a solution to the described problem of planning and decision making from which plants to pick chicks for processing by developing a web-based system which will provide decision support to the decision maker of the picking plan. Through the interaction with the system the decision maker will benefit and acquire hidden knowledge which will make them take the decisions which is more effective in terms of cost and quality. Further than this the system will provide to the decision maker analytical information summarized and interpretable in visualization components to help the user to understand the data.

The system will have a web interface where the user will interact with it and take suggestions for plants that meet the policy requirements for picking. For that purpose, the user will have access to historical performance data of the of plant to make comparisons on performance data. The idea is to integrate a reasoning engine into the system where it will follow a suggested strategy to find plants that meet the performance requirements and quantity and weight constraints given by the sales team.

The purpose of the system is to provide decision support to the person who creates the picking plan and suggest plants based on the policy and behavior that the user coach the reasoning engine. For that purpose, from historical and forecasted data the system will construct the context of the reasoning instance and the rules will define the picking policy with some priority between them. Priority is useful because there are situations where the user may need to define rules that are stronger than others for example the required strategy described by the user may need to find the plant with the biggest quantity available for picking. In that situation the user is allowed to define for example a rule that says, “if a plant satisfies the demand quantity 100%”, then this rule has priority over the rule that says, “if a plant satisfies the demand quantity 80%” and if a plant satisfies the first rule, then will be appeared first on the suggestion. Forecasted data will be added to the context to give the ability the policy writer to have a logic which use the predicted data in the reasoning context and define rules which suggest or not plants based on the future state of performance. For example, if the forecast show that the FCR of the plant will be increased in the next few days the policy writer can define a policy to pick immediately these chickens in order to avoid the forecasted inefficient growing performance to happen which may lead to the increase of the feeding cost and obviously the overall growing cost. The user will have the ability to create and run multiple solutions to the system. With the term solution we mean a new policy with a set of rules. The concept is to have the user to enter the flock actual data for each plant for a specific day, usually the current day and then to define the target demand in terms of quantity and weight range. After that the user needs to define a solution or a policy that is carried out when the user asks for suggestions. The policy can be modified at any time and stored permanently to the system along with the context and the reasoning results of the solution. We want to provide to the user analytical visualizations which explains visually the data. The main goal through the interaction with user, is to see the system provide suggestions to the user which can be understood easily through the visualization charts and data tables and make the user verify the suggestions made by the system and come to the point to make modifications and improvements to the policy and acknowledge inside the presented suggestions adaptation to it's given instructions.

Architecture

In this section we provide a description about the architecture of the system. The system has a three-tier architecture with the presentation layer, application layer and the data layer. On the data layer we use MongoDB as database engine in order to store the historical data from the plants, the different solutions given by the user, inference data, the latest context of the solutions and forecasted data. The data stored in structured format and the database is consumed by the various backends in the application layer. In the application layer we have the prolog backend where the prudens reasoning engine is running, the python tornado backend which is responsible to make the predictions and finally the C# backend which orchestrates all the others. In the presentation layer we have the web-based app which is written in angular and served by the asp.net backend.

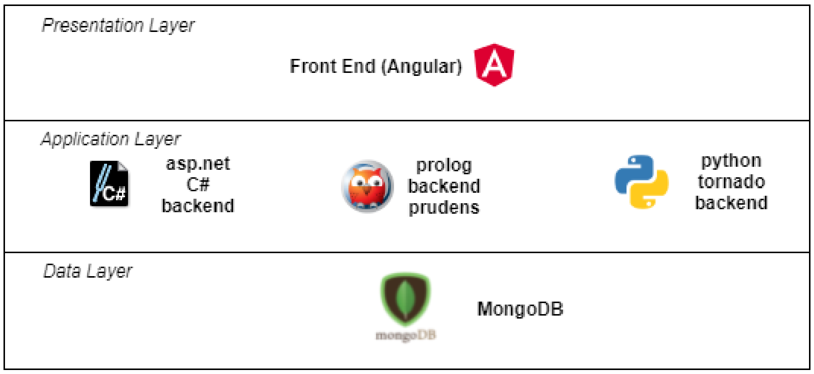


FIGURE 29 SYSTEM ARCHITECTURE

Presentation Layer

Web application - Angular

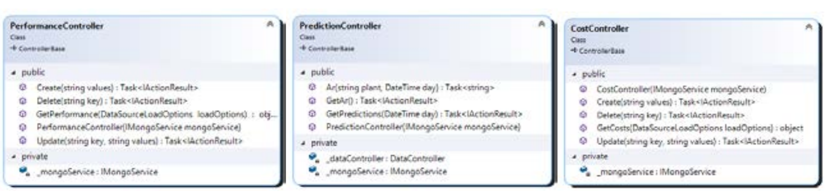
The presentation layer runs on the client browser which renders the web application to the end user. The frontend developed with the use of Angular, a modern web framework which is maintained by the angular team in google [47]. We use this framework to develop the graphical user interface with the use of community libraries to build up the whole user interface. The communication with the backend is done with the use of webservices which are called from the frontend and get the response from the backend. The whole web application is served by the asp.net backend and all the webservice calls target the asp.net backend.

Application Layer

The application layer contains the various backends needed to handle the application logic of the system. In this layer we have three different backends.

Orchestrator - Asp.net

The asp.net backend is the orchestrator of the whole system. Is responsible to serve the angular application acting as webserver of the front-end application, handles all the requests from the front-end through the http API and integrates the predictor backend, the mongo database and the prolog backend. The application is written in asp.net core using the net5 framework from Microsoft in C# language. The code is written in object-oriented fashion and reuse various packages to build up the application. In the figures bellow we show the class diagrams with high-level API description. The application is divided to multiple API controllers which handles the web request made by the front-end application and also makes requests and getting responses from other systems. On the Process Controller the PrudensProlog method handles the integration with the prolog backend. It accepts as method parameter the solution object which contains the context and the policy the need to be solved. We create background command line process and execute the command swipl init.pl which runs the init.pl file using the swipl.exe.



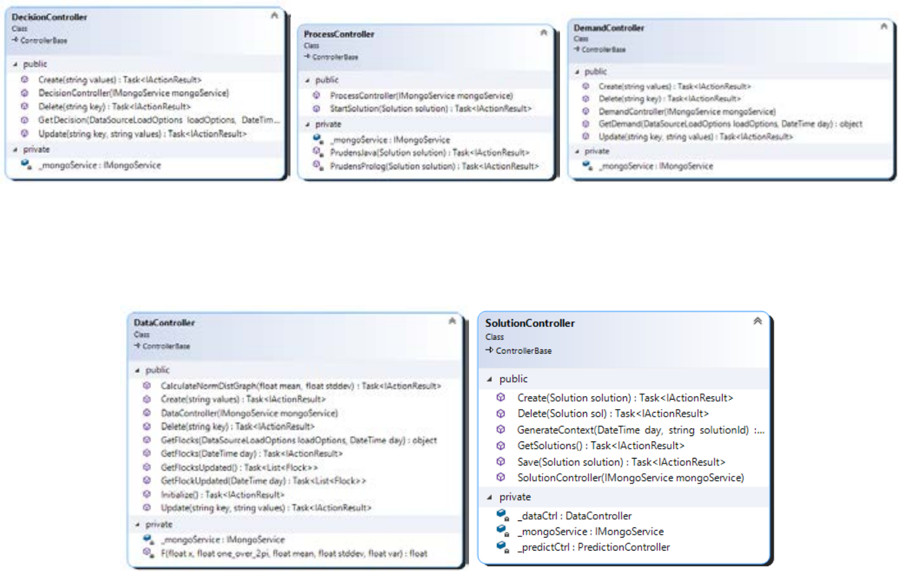


FIGURE 30 CLASS DIAGRAMS OF THE ORCHESTRATOR BACKEND

Below is the prolog template code which is contained on the init.pl file. In this template code when a new solution needs to execute, we create on the fly the initialization file and replace the strings <kb> with the policy rules and the <context> with the context facts. The reasonSRV is a higher-level service of the prudens engine in is getting called on the initialization. We redirect the standard output and standard error of the command line shell and we get the output of the prolog application. Finally, we extract the solutions and the elaboration information from the plaint output result.

The interaction with prediction service is done by the prediction controller from the method Ar. The flock data are passed to the python backend, and we get on the response a five-day forecast.

```
:- initialization main.
main :-
    working_directory(_, '<dir>'),
    consult('service.pl'),
    reasonSRV('<kb>','<kb>',[<context>]),
    halt().
```

FIGURE 31

Prudens - logic reasoning engine - Prolog backend

For the reasoning backend we experiment with prudens advice taker [48] with two implementations which the first was in java [49] and the second in prolog [48]. We decided to use the later in prolog because it supports more operators in the rule construction process. The idea of the advice taker follows an elaboration cycle, and in each cycle, elaborates information from a given context by making inferences from the rule-based predefined knowledge base. The elaborations may contain indications, actions in conjunction with an explanation with analytical steps showing how it reaches to the particular conclusion. The advice giver offers the advice to the advice taker and the advice is added to its knowledge base and used by the next elaboration cycles. The orchestrator backend is responsible to provide the contexts and the knowledge bases the reasoning system which in turn will execute the elaboration procedure and respond to the orchestrator with a report of the elaborated information.

Predictor - python tornado

The python backend is a micro-service which acts as a webserver and handle the tasks of forecasting. We use the tornado python web framework [50] to expose the services of the backend through http protocol. The goal is to construct an autoregressive model with the use of the python library statsmodels [51]. Basically, when a prediction needed the asp.net backend will call a webservice from the python backend and pass all the data needed. Usually we predict the FCR, the weight and the coefficient variation for the next five days. The asp.net pass all the historical data for a plant which retrieved from the mongo database so the python backend it doesn't need to query the database. Then we construct an autoregressive model from the training data and finally we get a forecast for the next five days on the input variables. Is a simple autoregressive model which is a flexible to handle different time series patterns. An autoregressive model can be seen as a multiple regression model with the difference that in autoregression we forecast the variable of interest using a linear combination of the past values (lagged values) of the variable instead the linear combination of predictors [52]. If the input data are not enough then the prediction is not possible to be done due to lack of historical data. For each plant we use the current cycle historical data, and we don't use any data from past cycles and this implies that the AR model doesn't capture any correlations of seasonality trends on the input variables.

Individual Visual Components of the system

In this section will provide a detailed presentation of the individual visual component of the web interface. In the figure fellow we illustrate the web interface of the system which is divided into two main parts. The taskbar and the main area.

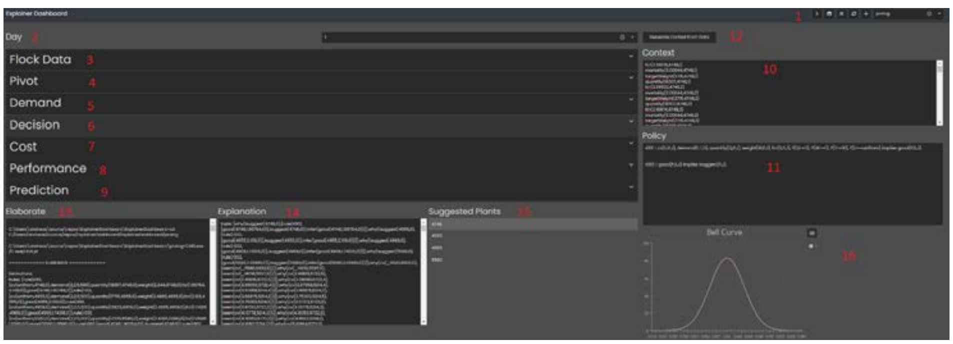


FIGURE 32. WEB INTERFACE OF THE SYSTEM

Taskbar

In the taskbar area the user can create, save and delete solution plans. Solution plans are individual plans made by the user on the same input of data but with different coaching approach. A single plan can be executed from the user by clicking the execution button on the task bar area. After the execution individual visual components will be appeared in the screen to provide support and suggestions to the end user.

FIGURE 33. TASKBAR

Day

The user can switch to the current and past days of the simulation. Every time the user switch day the data of the selected day will be displayed and also the context of the day will be generated. The day can be expressed as regular date time and must not be confused with the growing day which indicates the growing day of a flock in a specific plant.

FIGURE 34. CURRENT DAY

Flock Data

In flock data table view the user can trace the plant historical data and get an overview about the growing performance of the flock. Performance metrics are data that are imported implicitly or explicitly to the system. The user can enter manually the performance data after a briefing from the stockman of the plant or the data can be acquired automatically from the monitoring device installed to the plant through the local network. In this table some metrics are calculated from other base metrics. Bellow we indicate the base and calculated metrics along with their description.

Base Metrics

- *Grow Day* – Indicates the growing day of the flock.
- *Plant* – Is the unique identifier of the plant.
- *CV* – Is the coefficient variation captured in a growing day. Is expressed as numeric number.
- *CV Class* – Is the coefficient variation class expressed as categorical attribute.
- *FCR* – Feed consumption rate
- *Mortality* – Is the percentage of mortal chicks of the whole chicken population in the current day.
- *Qty* – Total quantity of chick in the current plant
- *Avg Weight* – Is the average weight of the growing flock. When is red then indicates that is below the target weight.
- *Feed Consumption* – Is the amount of feed consumed by the flock in a day and is expressed in KG.
- *Target Feed Consumption* – Is the potential amount of feed that will be consumed by the flock in a specific day and is expressed in KG.

Calculated Metrics

- *CV Avg* – Is the average coefficient variation until the current growing day.
- *CV Class* – Is the current coefficient variation expressed as a categorical attribute and is calculated based on the table in figure 15.
- *Target FCR* – Target Feed consumption rate from performance cards

- *Total Feed Consumption* – Is the total amount of feed consumed by the flock from day 1 and is expressed in KG.
- *Cost* – Is the total cost for the current growing day. Is calculated based on the current day feed consumption and with the conjunction of the cost table which indicates the cost of feed per KG on a growing day range.
- *Avg Mortality* – Is the average percentage of mortal chicks of the whole chicken population including the current and all previous growing days.
- *Target Weight* – Is the target weight on the current growing day. Is calculated from the performance table which shows the weight on a given day
- *Total Feed Consumption* – Is the total amount of feed consumed by the flock in until current day and is expressed in KG.
- *Cost* – Is the total cost of feed that consumed by flock in current day.
- *Cost per bird* - Is the total cost of feed that consumed by flock in current day divided by the population of the flock.
- *Total Cost per bird* - Is the running total cost of feed consumed by flock until current day.

Flock data view shows a table with flock all the available performance metrics. The user is able to manually add and remove rows and also modify the records. Also is able to apply filter on the entries and sort columns. This is the main view which represents all the available data for discovery.

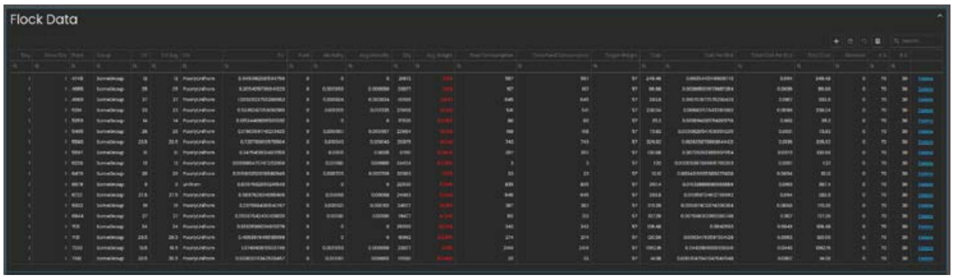


FIGURE 35 FLOCK DATA VIEW

Pivot

In the pivot area the user can navigate to see more details on historical data and can go and see information from the past days. In addition, the pivot area can visualize the aggregated data in a bar chart form at the top. This idea is coming from Business Intelligence systems where in multiple situations the user has the ability to filter and make aggregations on the data to get a representation of data in structured form with focus on the aspects that the user is searching for.

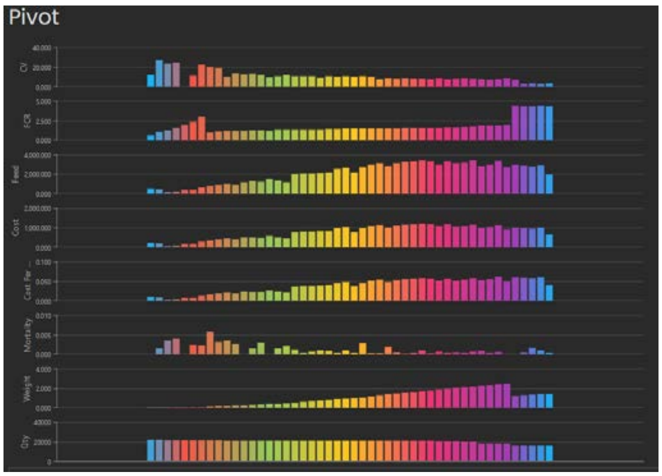


FIGURE 36 PIVOT CHART WITH MULTIPLE DAYS FOR A SINGLE PLANT

When the user hit on the execute button the pivot date area is displaying only the suggested plants from the results of the inference. Finally multiple plants and days can be chosen from the filter to get a bigger picture and compare analytics among the plants.

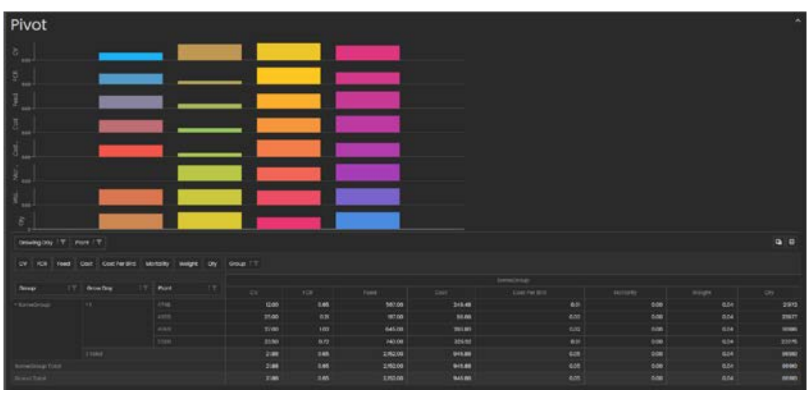


FIGURE 37 PIVOT TABLE VIEW FOR SINGLE DAY

Demand

In the demand view the user can write the desired demand of the selected day. It needs to specify multiple or single weight ranges (e.g., 1.20kg – 1.70kg) and the quantity needed. At this point we capture the requirements of stock needed at specific day and this information will be added in the KB context.

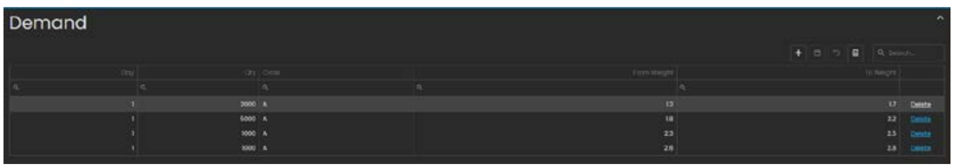


FIGURE 38 DEMAND VIEW

Decision

In the decision area the user can enter the actual decision taken by the decision maker. This information at this point is only a historical information for the actual decision which is not taken into consideration by the system and is not included in the KB context. It used to only for the evaluation purposes of the proposed solution to compare the actual decision taken compared to the alternatives proposed by the system. The user simply needs to specify the plant and the quantity removed from that plant.

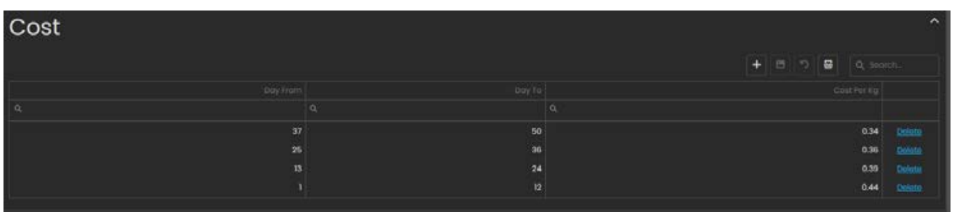
Decision

Date	Plant	Qty	
4/14/2021	4748	2000	Delete
4/14/2021	4955	3000	Delete
4/14/2021	4969	1500	Delete

FIGURE 39 DECISION VIEW

Cost

In this view the user enters the cost of feed per kilo in multiple day ranges. As we mention earlier the growing process requires specific nutrition at different ages of the flock. Thus, the cost of feed differs because different recipes are used. Cost data used to calculate the feeding cost of the flock for planning and evaluation purposes. Later, on the evaluation phase of the system we will discuss in detail for the cost variable and how it's affected our evaluation.



Cost

Day From	Day To	Cost Per Kg	
37	50	0.34	Delete
25	36	0.36	Delete
13	24	0.39	Delete
1	12	0.44	Delete

FIGURE 40 COST VIEW

Performance

Breed suppliers usually provide to their customers performance cards to help the stockman to track the growing performance of the flock. For this purpose, the user needs to add or adjust the information of the current breed used performance card into the system and the information is added into the context KB. The user needs to enter the day the weight, the feed consumed by one bird and the FCR. Also, the performance card data are used during construction of the prediction charts and the information can be identified on the blue line which is the target value.

Day	Weight(g)	Feed	FCR	
0	42	0	0	Delete
1	57	13	0.23	Delete
2	73	17	0.41	Delete
3	91	20	0.56	Delete
4	111	23	0.66	Delete
5	134	27	0.75	Delete
6	160	31	0.82	Delete
7	189	35	0.88	Delete
8	220	39	0.93	Delete
9	256	43	0.97	Delete
10	294	48	1	Delete
11	336	53	1.04	Delete
12	381	58	1.07	Delete

Figure 41 Performance View

Prediction

When the context is generated, the user is able to see prediction data for a selected plant. In the figures bellow we show some examples of graphs. The variables which can have predicted data are the FCR the Weight and the coefficient variation. We illustrate a time series graph with the day on the X-axis and on the Y-Axis the value of the presented variable.

The points with green color we can distinguish the historical data before the selected day. With blue color we show the points of the target value which queried from the performance data stored in the database. The red points show the forecasted values and usually shows the next five days from the current selected date. We exclude the target value on the coefficient variation because we do not have available performance data for this metric. From the these graphs the user can make the comparisons between the target, actual and predicted value and get a visual verification about the suggestions made by the system based on the rules that have been provided to the reasoning engine.

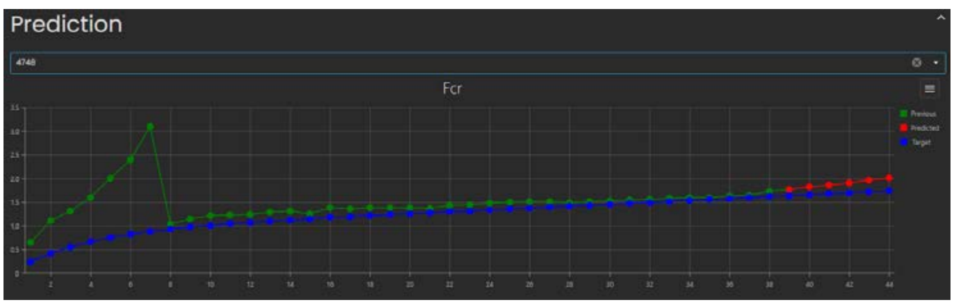


Figure 42 Prediction chart for FCR

Figure 43 Prediction chart for CV

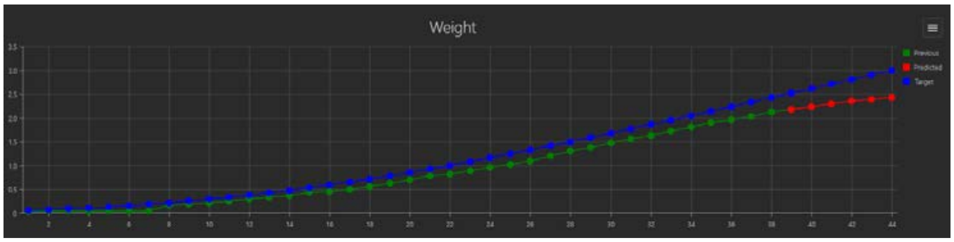


FIGURE 44 PREDICTION CHART FOR WEIGHT

Context

The context view can be described as the contextual knowledge of the system. The knowledge base contains facts which are taken into consideration in the inference process made by the system. The context data are passed to the prudens prolog system along with the policy. More precisely, facts are automatically generated in the context when the button “Generate context from data” is clicked. During the process facts are extracted from plants historical data and transformed into human readable facts. The context may contain actual or predicted data. Actual data are historical facts and predicted are facts which are generated by the predictors of the system for future days. In figure bellow there is an example of a context. The facts are written in a form of declarative prolog syntax. For example, the first fact in the example bellow is for the FCR of the flock which is around 2.01 in plant 4748 for day 1 (current day + 1). When day is specified as 0 this fact is from the current selected day. If day is bigger than 0 then the fact is forecasted.

FIGURE 45 CONTEXT VIEW

Policy

Policy can be described as the rules written by the user to describe the picking strategy. The syntax to construct a rule is very simple for a user or a programmer and it only needs to define the rules which contains variables from facts. With the “suggest” predicate the user indicates that a plant will be suggested when the rule is satisfied. Also, on the suggest predicate the user can define priority of the suggestion. Priority definition helps the user to define priority on rules over others and this indication is taken into consideration on the visualization part of the suggested plants where those are sorted based on their priority.

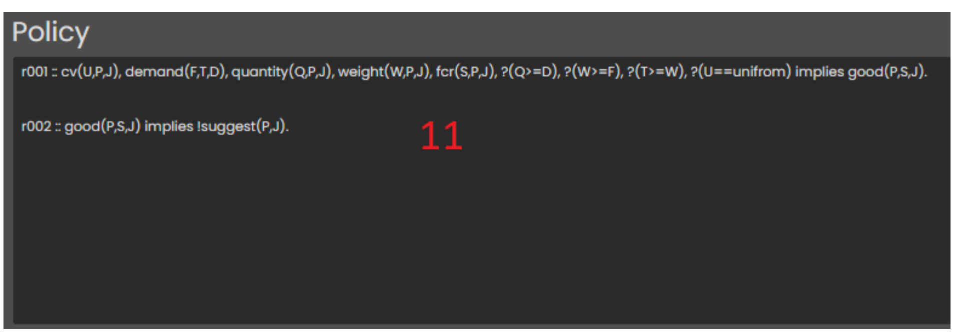


FIGURE 46 POLICY VIEW

Generate Context

As mentioned earlier the "Generate Context from Data" button when clicked is generating facts in the context view. We summarize the steps taken from this action:

- Fetch data from the database. First the system fetching the flock data of active plants for the current selected day. Then is fetching the demand and solution data.
- All the related data are stored in the memory of the system and then for each plant variables constructed and added to the context for the current selected day.
- For each plant we make a request to the prediction system and passing all the necessary information in order the python backend to make the prediction. We get the result, and we add the variable to the context. Each forecasted fact can be identified from the last parameter on the right. If the number is bigger than 0 then this indicates that this fact is forecasted.
- During the process some data go through normalization process or transformation from numbers to categorical attributes.

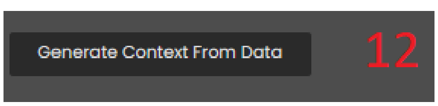


FIGURE 47 GENERATE CONTEXT FROM DATA BUTTON

Elaborate

In the elaborate view we show the detailed results coming from the prudens reasoning engine. It provides the suggestions made by the system along with the analytical explanation why the reasoning engine comes to that conclusion providing all the arguments taken into consideration during the process.

FIGURE 48 ELABORATE VIEW

Explanation

In the explanation view we get in detailed fashion the rules which are satisfied and which not, along with an explanation with the fired variables and conditions. This information is extracted after the execution of the prolog program and is part of the elaboration info.

FIGURE 49 EXPLANATION VIEW

Suggested Plants

When the execute button clicked if suggestions made by the system, they presented in the suggested plants view. The list is sorted based on the priority of the triggered rule which defined by the user in the policy area. Thus, the user can see priority on the suggested plants. When a user selects a plant the bell curve of the plant appeared. Also, there is a small explanation in parenthesis which passed by the satisfied rules.

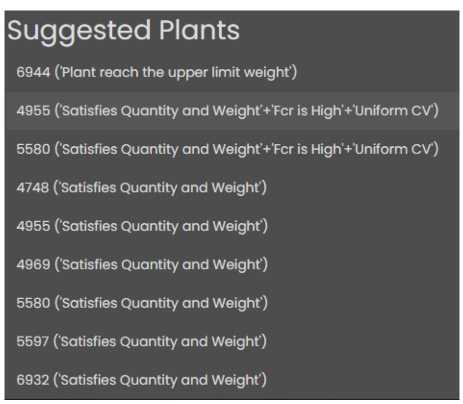


FIGURE 50 SUGGESTED PLANTS

Bell Curve

When the user clicks on the suggested plant a bell curve graph is displayed and visualize to the user the flock uniformity. The user can easily see the percentage of the total population for a lookup weight and reason the chances to catch the target weights from the given demand. To create the graph, we pass the weight mean and the standard deviation calculated by from the coefficient variation and mean weight, and we calculate the curve using a normal distribution.

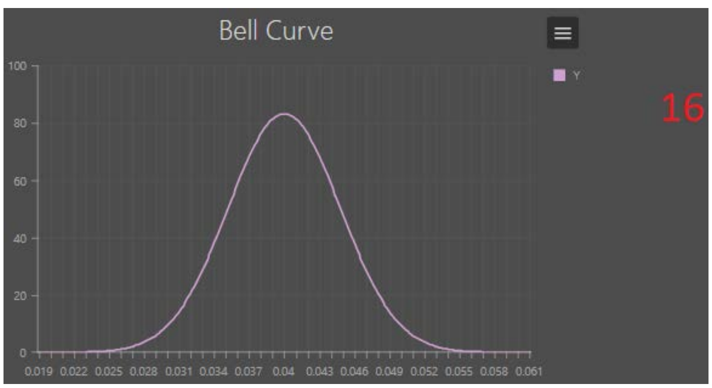


FIGURE 51 BELL CURVE

Chapter 4
Evaluation

For the evaluation of the system, we looked in the local industry of Cyprus to find companies that were interested to test and evaluate our solution, and we found a local company that is specialized in the poultry sector. The people there were friendly and interested to see the proposed system. We get very valuable feedback inside the discussions we made with the people involved. First, they explain to us how their production pipeline operates, and then they show us more analytical information about their infrastructure. Then they clarify which persons involved in the production plaining process and they go deeper into details of where they get the information about the growing performance of the flock and how they process the information listing all the reasoning steps until to come to the final decision. They point out also that in their business process except of their own poultry plants they have also external partners which are also part of the production planning loop and the same information like in the owned plants is evaluated from the external plants.

The company owning fifty plants and trading stocks from thirty plants belonging to external partners. Is good to mention here that in normal cases the plants are not always in operation since there are welfare rules that need to be followed and the plants may remain closed for a certain amount until the next cycle. Usually, a plant can have five to six cycles in a year and each cycle may take up to 46 days.

Picking Strategy

During the discussion with involved people, we come to the conclusion on the conditions that must be satisfied for a plant to be a good candidate for picking stock.

- Take from plants where their weights are close to the upper limit. (Upper limit can be defined as empirical fact from the user.)

  Reasoning: When the weight comes close to the upper limit e.g., the upper limit is 2.8 Kilos we need to process those chicks as soon as possible because we know after that limit, feeding can be translated as waste of money when the chicks enter a low growing phase, and the meat becomes harder.

- If mortality rate is high, then consider picking those chicks first even if not fitting the weight constraints (high mortality rate can be defined as a percentage empirically from the user)

  Reasoning: When mortality is high there is a high risk of losing the flock, so you need to process those chicks as soon as possible and reduce the damage cost.

- If FCR is high do not pick up except if you have not any other alternatives

  Reasoning: If the FCR is high this is a good indication that the flock is growing very efficiently so we can keep it growing with lower cost to reach higher weights ranges

- If FCR is low, then consider pick up those flocks as soon as possible.

  Reasoning:

  If the FCR is low, then it means that the flock is not growing very efficiently so we have to process them to avoid increase of cost in our production.

- Consider taking from plants with higher uniformity.

  Reasoning:

  Higher uniformity means better chances to target your weight constraints. Weights of chicks in a flock may have huge amount of variation and spread out of a wider range value. In this case we have less chances to target the weight. In cases of high deviation, we need to consider increasing the target quantity in order in order to have more chances to target the weight range and quantity. For example, if weight the range is 2.2 and quantity 2000 and deviation is high then is better to take 2200 exceeding the quantity constraints in order to reach the target weight range.

- Consider taking from plants that fit the weight, class and quantity constraints.

  Reasoning:

  You should always look for plants that meets the weight range and quantity and class constraints.

- Try to take from plants in the same group to reduce travelling distance and also keep in mind also the distance to the processing house and find the shortest path (the user can set a limit on number of different locations that can be visited)

  Reasoning:

  Wes needs to create an effective delivery plan. Try to avoid for example traveling in three long distance different places.

Evaluation Strategy

We decided for the evaluation purposes to involve two persons in the process. The “planner” which is the person who is the expert in the company and makes the picking plan, and the “programmer” which is the subject matter expert of the proposed system and has the technical background. We decided to use two persons in the loop because after an initial presentation of the system to involved people we come to the conclusion that the planner doesn’t have enough programming expertise to write the policy rules, thus we decide to involve the programmer which will interpret the rules in natural language from the planner and write the rules to the policy view.

We decided to evaluate the system in terms of production cost comparing the actual decision of the planner without the use of our system in contrary with the decision made by the planner after the suggestions made by the system, the validity of the suggestions made by the system and finally the feedback gained by the planner which is more like a questionnaire.

For evaluation purposes the data that will be provided to the system will be from historical performance data and decisions from the actual production planning process made in the past days. We decided to use past data than current fay data or near to today so to eliminate the chances of the planner memorizing its decisions. These steps of the evaluation methodology are the following:

1. A person involved in the planning process writes to a piece of paper decisions taken in the past involving only ten plants in the reasoning process. In order to hide the data from the planner we ask a different person to write the information. These decisions are the actual decisions that have been taken in the past. Then the planner gives to the evaluator the past performance metrics for each plant on the decision day in order the evaluator to calculate the feeding cost. The panner needs to give its decision for five past days of his choice with the only restriction that these days must belong to the same week. Essentially, we need historical performance and decision data for a past week.

2. The evaluator creates a table with the calculated total feeding cost for each day which will be used later to make comparisons.

3. The planner enters the historical performance data to the system for each plant and for each day. Also, the planner needs to add the feeding cost for the selected week. Then he communicates with the programmer about the first rule that need to be

added in the policy view. The programmer interprets the policy in natural language form and constructs the rule in the policy view. Is good to mention here that the planner has been initially informed by the evaluator to give to the programmer policies that are not complex at the beginning and the complexity of the rules will be increased in an incremental way. This strategy is useful for the evaluation purposes to see how the planner is responding on simple rules and how on more complex ones. We need to evaluate the behavior of the user after a suggestion have been proposed by the system and ask him if the suggestion make sense to him or not and why. Thus, we can evaluate if the user agrees with the suggestions draw by the system with a level of complexity.

4. The planner needs to provide its final decision for the selected day and gives to evaluator feedback about how it comes to that conclusion. Also, the planner may argue about the system proposals and provide its alternatives. Also, the planner has been informed to give detailed answers indicating system components which have been taken into its consideration to draw a conclusion.

5. For each decision we calculate the feeding cost and compare it with the actual decision in order to check if there is an increase or decrease on the production cost when different plants are chosen.

6. The programmer adds a new rule, and the steps of step four is repeated. At this point we increase the complexity of our policy and wait for the planner to give as a more detailed feedback since the policy is more complex.

The complete set of rules will be the following:

 # check if enough quantity

 # check weight range

 # for current day

r001 :: demand(F,T,D), quantity(Q,P,0), weight(W,P,J), ?(Q>=D), ?(W>=F), ?(T>=W) implies qtyOk(P,'Satisfies Quantity and Weight').

 # check if plants exceed upper limit weight

r002 :: weight(W,P,0), ?(W>=2.7) implies upper(P,'Plant reach the upper limit weight').

 # check if FCR is high

r003 :: fcr(W,P,0), ?(W>1.6) implies highFcr(P,'Fcr is High').

 # check uniformity to be uniform

r004 :: cv(U,P,0), ?(U<8) implies isUniform(P,'Uniform CV').

 # check if FCR will increase in the future

r005 :: fcr(C,P,0), fcr(F,P,1), ?(C<F) implies fcrInc(P,'Fcr increase in future').

 # suggest only plants with upper limit weight

r006 :: upper(P,C) implies !suggest(P,C,1).

 # suggest only plants which satisfy quantity, fcr and uniformity

r007 :: qtyOk(P,C), highFcr(P,F), isUniform(P,U) implies !suggest(P,C+F+U,2).

 # suggest only plants with quantity and weight satisfaction

r008 :: qtyOk(P,C) implies !suggest(P,C,3).

 # suggest plant which the fcr will increase in the future

r009 :: fcrInc(P,C) implies !suggest(P,C,4).

The last four rules are higher level rules and contain conjunctions of lower-level rules. In the suggest predicate the first variable is the plant, the second is the explanation defined in a lower-level rule and the third variable is the priority of the rule. For example, the rule r003 is lower-level rule which check if the FCR is high, and an explanation text was added into the implication predicate. As we mention earlier in the evaluation phase the higher-level rules will be added on by one.

Findings

In this section we provide the evaluation results and then we elaborate on the findings. In the table below we show the findings of the experiment after four attempts. For each day we start with low complexity rules and increase the complexity in each step. The planner needs to give four decisions for each complexity level and elaborate all the reasoning steps. In the table we record the decision taken before the experiment and the decision taken after the experiment. Also, we record the total cost per bird before the experiment and after. The cost is a critical factor to make comparisons between the decisions taken.

Is good to mention that in some cases if the cost of a decision is higher than other decision in some cases may be a good choice to choose the higher cost because in general, we suspend an expensive flock or a flock that is underperforming.

1. In the first attempt the user gets only one suggestion for plant 6944. This happening because the rule that is triggered is the r005 for the upper weight limit. The decision before was the plants 4748,4955,4969 with an average cost of 1.60. The first thought was to navigate to the flock data component to see all the plants and sort all the plants by their weight. The first response to the evaluator it was to point out that he understands the suggestion from the descriptive explanation in the suggestion list and he verified that after the sorting action on the flock data. The result shows that

No	Day	Rule Complexity level	Decision Before	Decision After	Total Cost Per Bird Before	Total Cost Per Bird After
1	18/04/21	1	4748,4955,4969	6944	1.60	1.71
2	18/04/21	2	4748,4955,4969	6944,4955,5580	1.60	1.72
3	18/04/21	3	4748,4955,4969	6944,4955,5580	1.60	1.72
4	18/04/21	4	4748,4955,4969	6944,4955,6932	1.60	1.75

this plant was over the limit of weight and it was one of the most expensive flocks. The decision it was instantly to pick up all chicks from this plant.

2. In the second attempt with complexity 2 the planner gets three suggestions from the system. Then navigates to the pivot component to see the summarized data with focus on the coefficient variation. Then he removed the filters from the pivot to list all the plants and he expanded also the bell curve graph and take a look on it. Next, he explained that the two newly suggested plants from the descriptive explanation foresee plants with high FCR with weights in range with high uniformity. The decision was to pick up from all the plants suggested, and he clarify that he preferred to take this decision because is more beneficial to pick up also from plants having good CV factor because is more likely to fall into the demand weight ranges.

3. At this attempt with complexity 3 the planner gets a list with seven suggestions. He navigated to the pivot component to see the performance data of the suggested plans. He identified that the newly suggested plants have low FCR, and the uniformity is moderate and in that case is better to keep the suggested flocks growing for a few days more. The decision was the same like in step 2. At this step the planner spends more time in the web interface to get support for his decision and also, he spend more time to respond to evaluator and give the final decision. He explained that the suggestion list was big enough, and he wanted to compare the suggestions spending more time on the analytical data and graphs.

4. At this step we added another one rule which will suggest plants that will have increased FCR in the future. The planner gets the suggestions list and focusses on the end of the list where he identified from the descriptive explanation plants that will have increased FCR in the future. Then he navigated to the prediction view and selected the suggested plants and analyzed the prediction graphs with more focus on the FCR. Also, he selected the plants that decided to choose in the previous step, and he pointed out that some other plants may have more increase in the FCR than the previous selection. He noted that there are a lot of plants bellow the target value and some predictions shows the FCR to take an important increase, and that makes him skeptical. He explained that he needs to see cost comparisons and also check the increase on cost in the next days. Then he navigated to pivot graph and looks on the cost graph. Then he moved to the flock data view and scrolled to the total cost column and identified that the plant 6932 is the most expensive flock compered to all the others. Have that in mind he selected the previous date from the date picker and noted down the total cost. Then he calculated the difference with the current day and found roughly that for each day the feeding cost is around 782.34. Then he does the same for some other plants and he found that if he chooses the plant 6932 may save around 250 for each growing day. Finally, he added that this step was the most interesting one and he emphasized on the prediction graphs and suggestions that support him enough to revise his previous decision. The final decision it was the same as the previous, but he replaced the plant 5580 with 6932.

Discussion

When the experiment finished, we asked the user to give us an overall feedback from his experience with the system. We divided the feedback given into two sections of advantages and disadvantages as mentioned by the user.

Advantages

- The suggested plants by the system after validation through interaction with the visual components of the system showed that suggestion have good reasons to appear in the suggestion list.
- Each time the complexity of the rules increased the entries in the suggestion list increased but the descriptive explanation it was more analytical and beneficial for the user to get more insights. The user needed more time to validate the results through interaction with the other component of the system, but this process was more beneficial in order to support his decisions.
- The system was flexible enough to accept changes on the policy and see an immediate adaptation of rules on the results.
- The predictions made by the system showed information to the user that wasn't able to count it using the traditional methodology of planning. Also, the calculation of cost it was an important indicator to look.

- The immediate access to all historical data of the plants in a structured minimalistic way it was one of the most beneficial functionalities of the system and having all the data gathered in centralized system increase the productivity of the planner and eliminate confusion issues experienced in the traditional methodology. The simplicity of the UI components supported the user to find the information he is looking for almost immediately without the need to open other documents or navigate to other systems to validate the information.

Disadvantages

- The system provides suggestions without indicating on the plant in which weight range of the demand is fit in. These make the user to go to the flock data view to cross reference information manually. This situation decreases the productivity of the user.

- The system suggests plants arbitrarily without suggesting how much quantity for each weight range to take out from the plant. Also, plants in the suggestion list may appear more than once with different descriptive information. The user preferred to see the plants distinct in the list with grouped descriptive explanation.

- The system does not take into consideration the distance of the plant from the processing house. The user expected by the system to make suggestions to take stock from plants that are in the same group to avoid multiple routes which will increase and travelling cost and human resources to accomplish the task.

- The user mention that the syntactical approach of the policy rules is difficult for a regular user to learn and it may take some time until to digest it. He pointed out that it will be easier if the policy can be defined using a visual logic component something similar to visual programming with diagrams.

- The system provides any number of suggestions and there is not such functionality to limit how many suggestions will be made. The user added that as the complexity increase the suggestion list becomes bigger and make him confused because it has multiple options and needs to validate the suggestions one by one and make him unproductive. He suggested to the evaluator to add a functionality to limit the maximum number of suggestions.

In the results table we noticed that the total cost per bird for each decision increased as the complexity level increased also. This fact made clear that when the user gets more descriptive information about the performance of each plant, takes decisions with plants that are expensive and that indicate that the user takes decisions which will have decrease on the overall cost since is removing earlier the flocks with higher cost. Furthermore, in the findings we saw that in each step the user revises his previous decision except in step three where the user chosen the same plants as before because as he explained he didn't get enough high-level descriptive information in order to revise his decision thus he decided to choose safely.

The advantages presented by the user shows that the most important functionality of the system is to provide to the user graphical and analytical components in order to validate the suggestions and support his thoughts until getting the final decision. Also, the prediction graph was a very important to be used in the last step of the experiment to see the future curves and derive proactive information to avoid underperformance of flocks. The forecasted facts in the context it was necessary for the system to draw suggestions which will alert and influence the user to analyze further what will be happened in the future and how this affecting the overall cost and performance and make the user skeptical enough to revise his traditional picking strategy with predictive methodologies. Also, the user mention that the suggestions rely a lot on the FCR metric which is a feeding performance metric. He explained that the yield metric is important to be added in the context in future releases of the system on in order to influence the system to make suggestion including the yield. Moreover, clarify that the rules which include the yield metric must have higher priority from the FCR because the yield shows the percentage of meat extracted from the body after processing and is a strong argument to be taken into consideration.

Conclusion

Future Steps

After the first implementation of the system and the evaluation experience in a real case scenario we have noted some aspects that may need future improvement and also new functionalities that need to be added to make the system richer providing more analytical and explanatory data. Moreover, on the problem definition we mention some aspects that missing from the implemented system and is good to be considered in future releases.

- In the problem definition we mention about the problem of planning picking stocks from plants for processing and also adding new flocks to the plants. The doesn't have the functionality to make suggestions in which plants to add new flocks. For that purpose, in the future, we need to add a visual component to the system which will have a Gantt chart like in figure 23. Also we need to add facts in the knowledgebase which indicates that a plant is available of placing new stock and the resting period has passed. Also is good to take into consideration adding flocks in plants that are in the same group so in the picking process to travel to only one location for picking. Additionally, the system needs to find from historical data a forecasted demand so the placing module of the system to count how much chicks will be needed for slaughtering on a future day. This will be beneficial to have a balance on the quantity of birds placed and avoid coming to situation to have less stock of the demand or more.

- The system needs to take into consideration the distance from the slaughtering house and make suggestions that do not exceed a limit of kilometers that need to be traveled. Furthermore, the system needs to find the most cost-effective routing plan and count also the human resources needed to accomplish the operation. For example, if the system suggests taking from 5 plants which are not in the same group and all the plants are in different location is obvious that this suggestion is inefficient because the planner needs to send five trucks and some human resources to five different location. If the suggestion is to take from three plant in the same group, then is clear that routing plan is more efficient since one truck is needed with few human resources.

- From the feedback we get from evaluation phase we understand that is important to include quality data from inspections made by the quality team during slaughtering process. Metrics like the yield of the flock, incidents that appear in the slaughtering process and classification of quality are very important data for the planner to take into consideration. For that purpose, we need to add an inspection view where those inspections will be recorded in a daily manner and include facts in the context of the reasoning engine and rules in the policy view related to inspection data.

- Provide an interface for construction of rules in diagrammatic approach to engage non-technical people to provide solutions using visual programming components. Also, simple build in aggregation functions like counts, max, min, avg, sum can be developed and used in the visual programing toolbox of the interface to provide to the policy designer aggregation formulas to write more complex rules.

- On the prediction component we need improvements on the current constructed models. We need to consider algorithms and training data from past flocks which make seasonality relations and find seasonality trends among the data.

We conclude this work saying that we are satisfied with the fact that the planner with cooperation with the programmer have accomplished the task of planning picking chicks for slaughtering and the result of the evaluation shows that the user takes decisions to eliminate under-performance stock and furthermore the user gets more insights about the flock performance in analytical and graphical manner. In addition, we identified some future steps to improve our work in order to provide a robust tool in the poultry industry to make an efficient picking plan in a productive way.

References

[1] A. Felsberger, "A Review of Decision Support Systems for Manufacturing," 2017.

[2] M. W. J. F. C. D. J. J. P. Shim, "Past, present, and future of decision support technology. Decision support systems," 2002.

[3] P. G. Keen, "Decision support systems: a research perspective. Decision Support Systems: Issues and Challenges," 1980.

[4] C. H. K. B. B. Tregoe, "The new rational manager: An updated edition for a new world," 1997.

[5] J. S. Dyer, "Maltiattribute utility theory (maut). In Multiple Criteria Decision Analysis," 2016.

[6] E. Turban, Aronson and T.-P. Liang, "Decision Support Systems and Intelligent Systems," 2005.

[7] D. J. Power and R. Sharda, "Model-driven decision support systems," 2007.

[8] Z. Pourabdollahi and B. Karimi, "Shipping chain choices in long-distance supply chains," 2014.

[9] I. Ivlev and J. Vacek, "Multi-criteria decision analysis for supporting the selection of medical devices under uncertainty," 2015.

[10] C. Saha and F. Aqlan, "A decision support system for real-time order management in a heterogeneous production enviroment," 2016.

[11] C. J. Hanlon and J. B. Stefik, "Statistical decision analysis for flight decision support," 2013.

[12] B. Li and J. Li, "Demand forecasting for production planning decision-making based on the new optimised fuzzy short time-series clustering," 2012.

[13] G. Figueira, "A decision support system for the operational production planning and scheduling of an integrated pulp and paper mill," 2015.

[14] J. R. Munhoz, "Optimization approaches to support decision making in the production planning of a citrus company," 2015.

[15] M. Yadollahpour, *A comprehensive solution for continuous casting production planning and scheduling.*

[16] R. Seethamraju, *Enterprise systems and demand chain management: a cross-sectional field study,* 2014.

[17] A. Salo, "Impact of the shape of demand distribution in decision models for operations management," 2013.

[18] L. Wang, "Dynamic emergency logistics planning: models and heuristic algorithm," 2016.

[19] R. A. Junqueira, "Production and logistics planning considering circulation taxes in a multi-plant seed corn company," 2012.

[20] C. Li , "A system dynamics simulation model of chemical supply chain transportation risk management systems," 2016.

[21] J. Rouzafzoon, "Developing service supply chains by using agent based simulation," 2016.

[22] D. J. Power, "Supporting decision-makers: An expanded framework.," 2001.

[23] D. J. Power, "Decision support systems: a historical overview," 2008.

[24] D. J. Power, "Understanding data-driven decision support systems," 2008.

[25] J. Daniel, "A brief history of decision support," 2007.

[26] R. Sindhgatta, "Using an information retrieval system to retrieve source code samples," 2006.

[27] G. Desanctis, "A foundation for the study of group decision support systems," 1987.

[28] D. DeTombe, "Group decision support systems," 2015.

[29] "Computer supported argumentation and collaborative decision making," 2001.

[30] V. Karan, "Information technology support for collaborative decision making in auditing," 1996.

[31] K. L. Kraemer, "Computer-based systems for cooperative work and group decision making," 1988.

[32] M. Negnevitsky, "Artificial intelligence: a guide to intelligent systems," 2005.

[33] H. R. Nemati, "Knowledge warehouse: an architectural integration of knowledge management integration of knowledge management support, artificial intelligence and data warehousing," 2002.

[34] A. Chakir, "A decision approach to select the best framework to treat an it problem by using multi-agent system and expert systems," 2016.

[35] A. Abraham, "Rule-based expert systems. Handbook of measuring system design," 2005.

[36] A. K. Kar, "A hybrid group decision support system for supplier selection using analytic hierarchy process, fuzzy set theory and neural network," 2015.

[37] D. Arnott, "A critical analysis of decision support systems research," 2005.

[38] L. Michael, "Machine Coaching," 2019.

[39] N. Blenn and F. Kuipers, "LoRaWAN in the Wild: Measurements from The Things Network," 2017.

[40] "Ross Broiler Management Handbook: Preface," 2018.

[41] M. D. I. D. Vale MM, "Characterization of Heat Waves Affecting," 2010.

[42] E. N. O. Akanmode, "Prediction of Poultry Yield Using Data Mining," 2018.

[43] E. A. Mehmet Mendes, "Regression tree analys for predicting slaughter weight in broilers," 2009.

[44] V. R. J. V. M. Moura DJ, "Heat stress impact on weight gain in broiler chickens," 2015.

[45] "Ross 308 Performance Objectives 2019," Ross, 2019.

[46] "Carcass and parts yield of broilers reared under a semi-extensive system," Vols. Faria PB, Bressan MC, Souza XR de, Rossato LVI, Botega LMG, p. , 2010.

[47] "angular.io," Google, 2021. [Online]. Available: https://angular.io.

[48] L. Michael, "The Advice Taker 2.0," 2017.

[49] V. T. Markos, "Application of the Machine Coaching Paradigm on Chess," 2020.

[50] "Tornado," FriendFeed, 2021. [Online]. Available: https://www.tornadoweb.org/.

[51] S. Seabold and Josef Perktold, "Statsmodels: Econometric and Statistical Modeling," 2010.

[52] G. A. Rob J Hyndman, Forecasting: Principles and Practice, 2013.

[53] W. A. Teniwut, "Decision support system in supply chain: A systematic literature review," 2020.

[54] M. Abdulllah, "A Review of Automated Decision Support System," 2018.