

Designing a Machine Learning – Based Framework for Enhancing Performance of Livestock Mobile Application System

Herbert Peter Wanga¹, Nasir Ghani², Khamisi Kalegele¹

¹School of Computational and Communication Sciences and Engineering (CoCSE), Nelson Mandela African Institution of Science and Technology (NM-AIST), Arusha, Tanzania

²College of Engineering, Department of Electrical Engineering, University of South Florida, Florida, USA

Email address:

wangah@nm-aist.ac.tz (H. P. Wanga), nghani@usf.edu (N. Ghani), khamisi.kalegele@nm-aist.ac.tz (K. Kalegele)

To cite this article:

Herbert Peter Wanga, Nasir Ghani, Khamisi Kalegele. Designing a Machine Learning – Based Framework for Enhancing Performance of Livestock Mobile Application System. *American Journal of Software Engineering and Applications*. Vol. 4, No. 3, 2015, pp. 56-64.
doi: 10.11648/j.ajsea.20150403.13

Abstract: Smallholder livestock keepers live in rural areas where there is poor Internet connectivity. Many mobile based system designed do not function well in such areas. To address these concerns, an Android Mobile Application will be designed and installed on a smartphone. The application will have an easy to use Graphical User Interface (GUI) and request resources from the server through the Internet. This Intelligent Livestock Information System (ILIS) will be able to provide and predict feedback to the livestock keepers. This solution will also collect livestock data from livestock keepers through mobile phones. The data will then be sent to the database if connectivity is available or through synchronization if connectivity is poor. Livestock experts will be able to view data and respond to any query from livestock keepers. The system will also be able to learn and predict the responses using machine learning techniques. The goal of the ILIS is to provide livestock services to anyone at anytime, overcoming the constraints of place, time and character. Overall, this is a novel idea in the field of mobile livestock information systems. Along these, this paper presents the software, hardware and architecture design of the machine learning based livestock information system. Overall this solution embodies an artificial intelligence approach which combines hardware and software technologies. The design will leverage the Android ADK operating system and Android mobile devices or tablets. Our main contribution here is the intelligent livestock Information System, which is a novel idea in the field of mobile livestock information systems.

Keywords: Intelligent Information System, Machine Learning, Android ODK, Requirements, Modeling, Artificial Intelligence, Livestock App

1. Introduction

With the rapid increase in population and livestock economic development, poor exchange of livestock information is a cause for increasing concern. Many livestock keepers live in rural areas, where there is poor connectivity. Hence the provision of an intelligent livestock information system in such poor connectivity areas is of vital importance, and this is the focus herein.

In our design, we will use smartphone devices, installed with a developed Android application. The device will have SQLite for data synchronization with a remote database. Machine learning techniques will also be deployed in the device. By using these devices, information flow will be

supported between livestock keepers, field officers, researchers, and district officers. The designed system will be also able to replicate from the remote database to the SQLite. Machine learning will be able to use experience and make predictions. We will use machine learning to program systems to automatically learn and to improve with experience, hence making intelligent decisions based on data. It will improve performance in some tasks using experience.

As per [1] machine learning addresses two interrelated problems, namely the development of software that improves automatically through experience and the extraction of rules from a large volume of specific data. Therefore systems capable of exhibiting such characteristics are important because they have the potential to reach higher levels of

performance versus other systems that must be modified manually to deal with situations their designers did not anticipate.

In the design stage, given the number of requirements on software complexity and quality, it is important to present detailed requirements on software structure, on navigation by software, on design of user interfaces, on multimedia components of software, on usability, and on technical requirements [2]. This study gathered all necessary requirements towards designing and development of the intelligent livestock information system.

The intelligent program can be implemented on the client's mobile for the purpose of smooth working within the application program, in a fully or partially disconnected environment [3].

When the mobile clients do not have synchronized data, they are dependent on servers for continuous support because of the intermittent nature of the mobile networks. Hence in such cases, mobile clients must be capable of operating in the disconnected mode for some periods of time, depending on the critical need of the synchronized data. This program implementation eliminates the other system, which assumes constant connectivity [4].

During disconnected operation, updates are persistently logged by the system, using a variety of optimizations to reduce resource consumption. For example, if a file is created, renamed, and later deleted, none of those records need to be saved; i.e., it is as if none of the operations ever happened. When connectivity is re-established, the program should replay these updates at the servers through reintegration [3].

As a large amount of livestock data is handled by the server, the server will perform mining and analyze data. Hence by performing analysis and by using suggestions and information provided by the livestock experts, the server can learn and provide feedback automatically. Moreover as time goes on the server will be trained automatically by mining and analyzing data of all the possible livestock practice scenarios and become more and more intelligent.

Our main contribution here is the intelligent livestock information system, which is a novel idea in the field of mobile livestock information systems.

The remaining part of this paper is organized as follows. Section 2 briefly describes some related applications. Next, Section 3 presents the analysis on the requirements of a machine – learning based framework for enhancing performance of livestock mobile application system, Whereas Section 4 overviews the design of the proposed system. Section 5 concludes the paper, and explains the future work.

2. Related Work

This section deals with some of the existing works related to the proposed mobile solution, and their challenges.

2.1. Existing Works

This section reviews some studies related to the machine-learning based framework for enhancing

performance of livestock mobile application system.

2.1.1. Livestock Gestation Tables

This app helps with breeding decisions for the larger farm animals. It provides a gestation chart for the 5 main species of farm animals, included access to a calendar. Hence users can determine when they want to breed animals to have offspring at the best time of the year.

2.1.2. Cow Tracker

If the cow is pregnant it will calculate the approximate due date. It has ability to see what events are coming up in the next 7 days. A user can also click on a cow within this view and it will take them to the cow's page. There is the ability to have notifications delivered for all three events as well and when the events will be delivered on a once per day basis. A heat check notification will arrive 3 days prior to the next heat date. A blood draw notification will arrive 5 days after the first time you can draw blood for testing. A calving notification will arrive 14 days prior to the estimated due date [5].

2.1.3. Mobile Cattle Tracker

While the majority of data on a beef cow-calf operation is collected seasonally at herd workings, certain data must be collected on an ongoing basis throughout the production year. Also, production data like birth records, weights and measurements and management activities needs to be entered real-time, so that it can be available for review when animals are gathered and worked. Mobile Cattle Tracker allows data to be collected in a consistent manner and on a timely basis so that meaningful information can be generated [6].

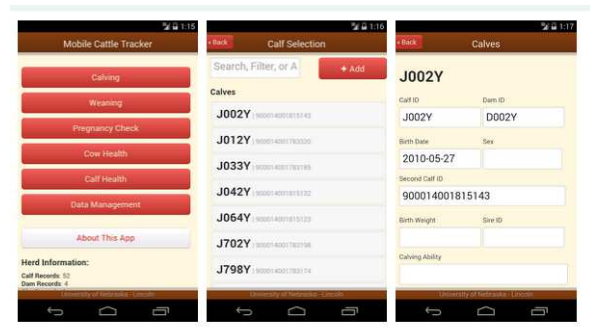


Figure 1. Mobile Cattle Tracker interfaces.

2.1.4. Willoughby Livestock

A one stop access to all divisions of Willoughby Sales through the convenience of a mobile device. This App makes it easy to interact with any of the live auctions. Features include: direct access to auction sites, GPS directions, detailed auction schedules, photo gallery, live forum and much more.

2.1.5. Livestock Feeding Made Easy

Authors in [7] say that this app allows one to enter the amount of livestock, e.g., such as dairy cows, beef cattle, cattle, and calves of all ages. Also information on sheep, deer and horses can also be entered. Then record the quality of the brass/silage or hay being fed can also be stored along with the amounts (if any) of concentrates being used. One can also find

out how much they need to use to stretch that feed. Finally, detailed reports can also be generated detailing the amount of livestock units, the amount they consume per day (kilograms) and how much concentrates they get.

2.1.6. Livestock Information Management System

As per reference [8], at the NBAGR, Karnal, a menu-driven livestock information management system was developed for animal resources. All the data tables and report forms are designed in the form of a single package. The database has information on livestock population, genetic resources, infrastructure, production, products and utilization, farms, etc. Separate tables have also been defined for each parameter. The master tables have names and codes of fields, which are repeatedly used in other tables, and there are data tables, which are linked to master tables and contain actual data.

2.1.7. Livestock Information Management System (LIMS)

The LIMS software package is designed to facilitate the recording and processing of livestock data in developing countries. The package is not specialized for use with any particular species or production system, but can be used with data for any mammal species for a variety of purposes, such as research, extension and commercial production. The objective is to provide users with a computerized tool that allows them to manage complex livestock performance data set, and to provide help in livestock management without relying on specialist computer and data management support.

The LIMS software package has been designed to address six key issues in management of livestock performance data, i.e., balance of flexibility and standardization of data definition, documentation and definition of data sets, assistance in deriving non-observed data (breed, parity, mating-parturition connection), data validation and error correction, reporting for animal management, and data extraction and calculation of standard performance traits for statistical analysis. In order to address these various aspects of livestock data management, the proposed LIMS software has a modular design, with different modules addressing each of the above mentioned issues. A generic data set is distributed along with the LIMS software. LIMS is developed in the CLIPPER language. All data in a LIMS data set are stored in dBASE III + compatible data files [9].

2.2. Challenges of Current Information Systems

Most of the applications mentioned above require continuous Internet access. Moreover, they cannot use experience to build upon prior knowledge and better respond to user requests. As such, they do not address the specific issues and concerns facing Arumeru district livestock keepers.

3. Requirements Analysis

In this section, we discuss the requirements analysis for designing of the machine – learning based framework for enhancing performance of livestock mobile application system. Now requirements analysis involves defining

customer needs and objectives in the context of planned customer use, environments, and identified system characteristics to determine requirements for system functions

System requirement is a need: a process or improvement that stakeholders want to realize through a system. It is a demand to a system: (functionality) or quality (performance) that a system must have to fulfill the need of the stakeholders [10].

3.1. Functional

Overall, the system should provide convenience and automation for both livestock keepers and livestock experts. It should also satisfy the following requirements:

- a) The system should provide plenty of informative instructions or guidelines to help livestock keepers find available modern livestock practices.
- b) The system should provide effective information to livestock experts and back to livestock keepers.
- c) The system should provide suitable marketing information.
- d) The system should provide powerful functions to help district livestock officers manage field livestock officers.
- e) The system should be able to work and collect information in poor connectivity areas.
- f) The system should be able to predict responses.
- g) System should be able to improve performance of livestock keepers in some tasks basing on experience.
- h) The systems should allow users to retrieve up-to-date livestock information.
- i) Intelligent processing using machine learning techniques.
- j) Developing system interfaces which can call and implement specific data processing in the system.

In accordance with the above requirements, the machine – learning based framework for enhancing performance of livestock mobile application system should require minimize human operation and supervision, so as to reduce the cost of manpower and the lost from human mistake and to enhance the extension services.

3.2. The Non-Functional Requirements Include

- a) Ease of use, so that developers can use the tool to their benefit;
- b) Ease of distribution and integration, making the tool available and ready for application;
- c) Extendibility, allowing developers to add functionality according to their specific needs.

3.3. Data Flow Diagram (DFD)

As per [11], a data flow diagram (DFD) graphically describes the flow of data within an organization. It is used to document existing systems and to plan and design new ones. Generally, there is no ideal way to develop a DFD. However it is composed of the following four basic elements: data sources and destinations, data flows, transformation processes, and data stores. In this study, the DFD shows how information is

delivered by the livestock keeper from various sources such as livestock researchers, field officers and livestock field officers.

3.3.1. Existing Data Flow Diagram

The existing model puts the livestock field officer at the middle between the livestock keeper and other actors. Hence all information passes through this individual.

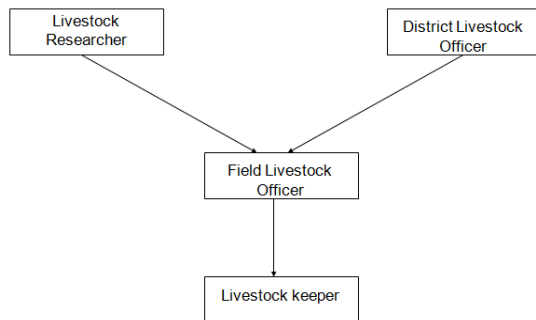


Figure 2. Existing information flow model.

Field livestock officers face challenges in dispersing useful information to livestock keepers who are geographically scattered. These officers generally do not have reliable means of transport and communication. Therefore there are barriers in conveying modern livestock practices and other extension services.

3.3.2. Proposed Data Flow Model

The proposed model will work in both internet connectivity and poor connectivity area basing upon machine learning techniques, see Figure 3 below.

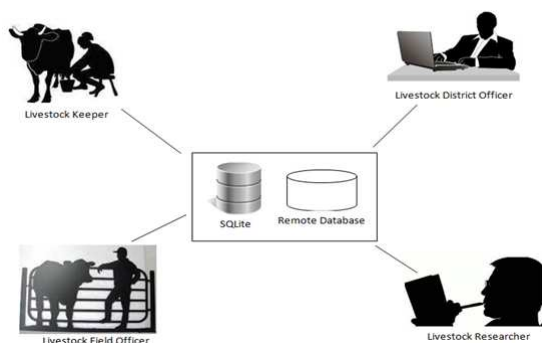


Figure 3. Proposed Data Flow Model.

The proposed data flow model has more advantages than the existing model. These include;

- Information is received by the livestock keeper directly from various stakeholders, as opposed to the existing one where data is received through the livestock field officers.
- Information is delivered through mobile phones.
- Field officers respond to queries on the system, not face to face.
- Researchers publish results on a central database which are accessible to every actor.
- System also works in poor connectivity areas.

- Machine learning responds to queries and can make predictions

3.4. Flowchart of the Information System

A flowchart is an analytical technique used to describe some aspect of an information system in a clear, concise, and logical manner. Flowcharts use a standard set of symbols to pictorially describe transaction processing procedures. Flowcharting symbols can be divided into the following four categories: input/output symbols, processing symbols, storage symbols, and flow and miscellaneous symbols [11]. Figure 4 below shows the flow chart of the proposed system.

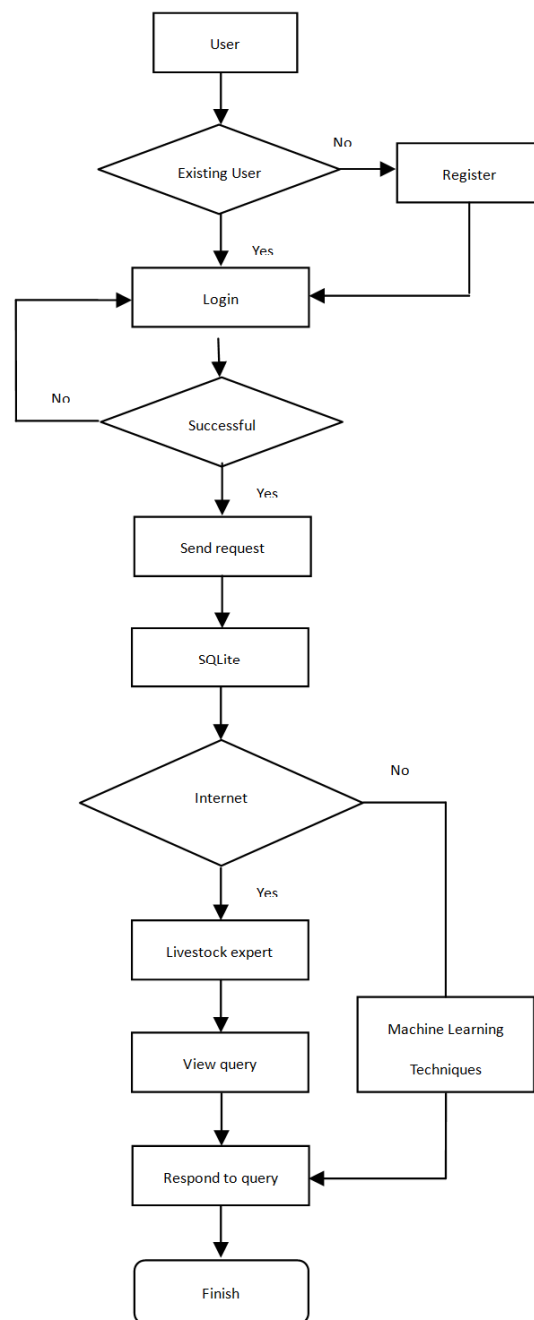


Figure 4. Flow chart diagram.

3.5. Differences Between DFDs and Flowcharts

According to [11], there are many differences between DFDs and flowcharts including:

- DFDs emphasize the flow of data and what is happening in a system, whereas a flowchart emphasizes the flow of documents or records containing data.
- A DFD represents the logical flow of data, whereas a flowchart represents the physical flow of data.
- Flowcharts are used primarily to document existing systems. DFDs, in contrast, are primarily used in the design of new systems and do not concern themselves with the physical devices used to process, store, and transform data.
- DFDs make use of only four symbols, whereas flowcharts use many symbols and thus can show more detail.

3.6. Use Case Diagram

A use case illustrates a unit of functionality provided by the system. The main purpose of the use case diagrams is to help development teams visualize the functional requirements of a system, including the relationship of "actors" (human beings who will interact with the system) to essential processes, as well as the relationships among different use cases [12].

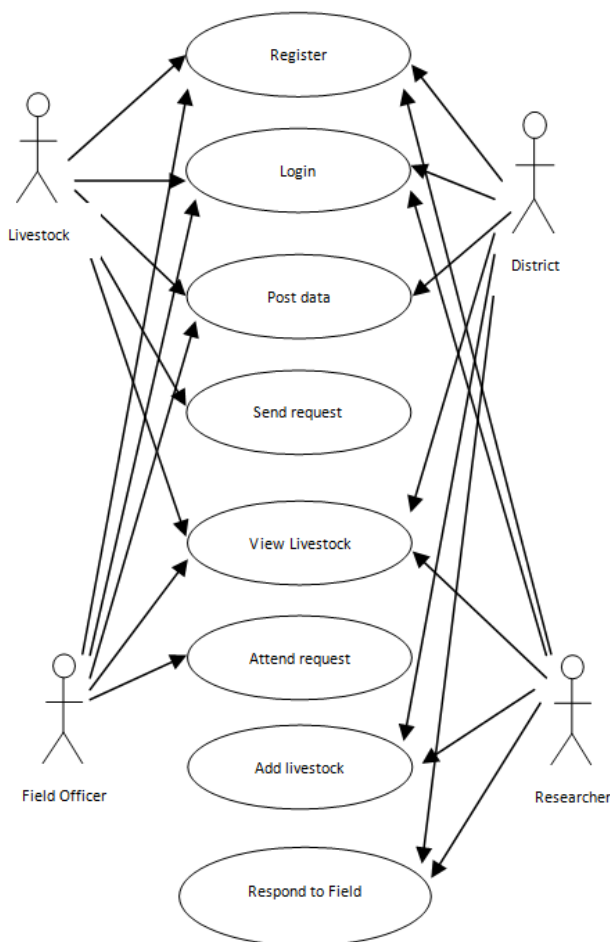


Figure 5. Information Flow Diagram.

3.7. Information Flow

An information flow model characterizes how information flows throughout an application by describing the kinds of processing that take place and how data flows between these stages.

In our study we will connect to a remote MySQL database via an Android application. Since these entities cannot communicate directly, we will use PHP. Namely, we will create a PHP-based web service that will allow our Android application to register a new user, login, and parse data within our application. Hence the design will need a server, a MySQL database, PHP knowledge, Eclipse, and the Android SDK.

3.8. Data Modeling

According to [13], data modeling is the process of learning about data, and the data model is the end result of the data modeling process. In addition, data modeling is a process used to define and analyze data requirements needed to support the business processes within the scope of corresponding information systems in organizations.

A data model organizes data elements and standardizes how the data elements relate to one another. As the name implies, the data model focuses on what data is required and how it should be organized rather than what operations will be performed on the data. The data model focuses on representing the data as the user sees it in the "real world".

According to [14], a data model is a way finding tool for both business and IT professionals, which uses a set of symbols and text to precisely explain a subset of real information to improve communication within the organization and thereby lead to a more flexible and stable application environment.

Data models are often used as an aid to communication between business people (defining the requirements for a computer system) and technical people (defining the design in response to those requirements). The data model can sometimes be referred to as data structure, especially in the context of programming languages. Data modeling comprises of Entity Relationship Diagrams (ERD) and Class Diagrams.

3.8.1. Entity Relationship Diagrams (ERD)

As per [15] an Entity-Relationship (ER) diagram provides a graphical model of the things that the organization deals with (entities) and how these things are related to one another (relationships). This approach presents a graphical representation of objects or concepts within an information system or organization and their relationship to one another. ER diagrams are typically used in computing to show the organization of data within databases or information systems. An entity is a piece of data, an object or concept about which data is stored. A relationship is how the data is shared between entities.

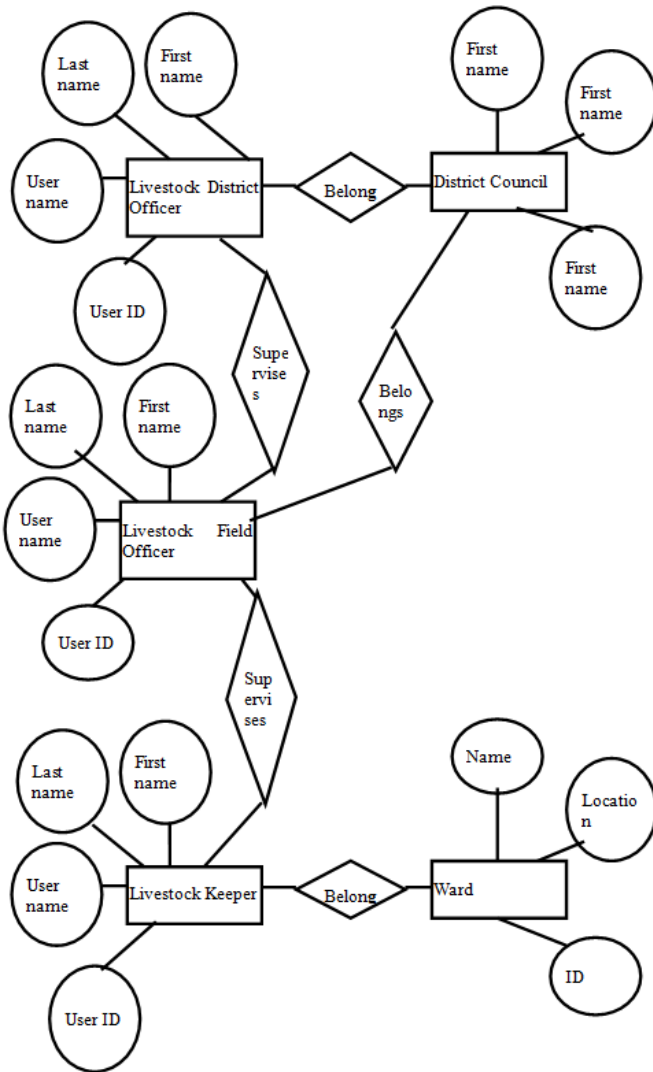


Figure 6. System E-R diagram.

E-R diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships, and ovals are used to represent attributes.

3.8.2. Class Diagram

A class diagram describes the types of object in the system and the various kinds of relationships among them. The UML representation of a class is a rectangle containing three compartments stacked vertically. The top compartment shows the class's name. The middle compartment lists the class's attributes. Finally, the bottom compartment lists the class's operations. Class diagrams describe the types of objects in the system and the various kinds of static relationships that exist among them. It shows how the different entities (people, things, and data) relate to each other.

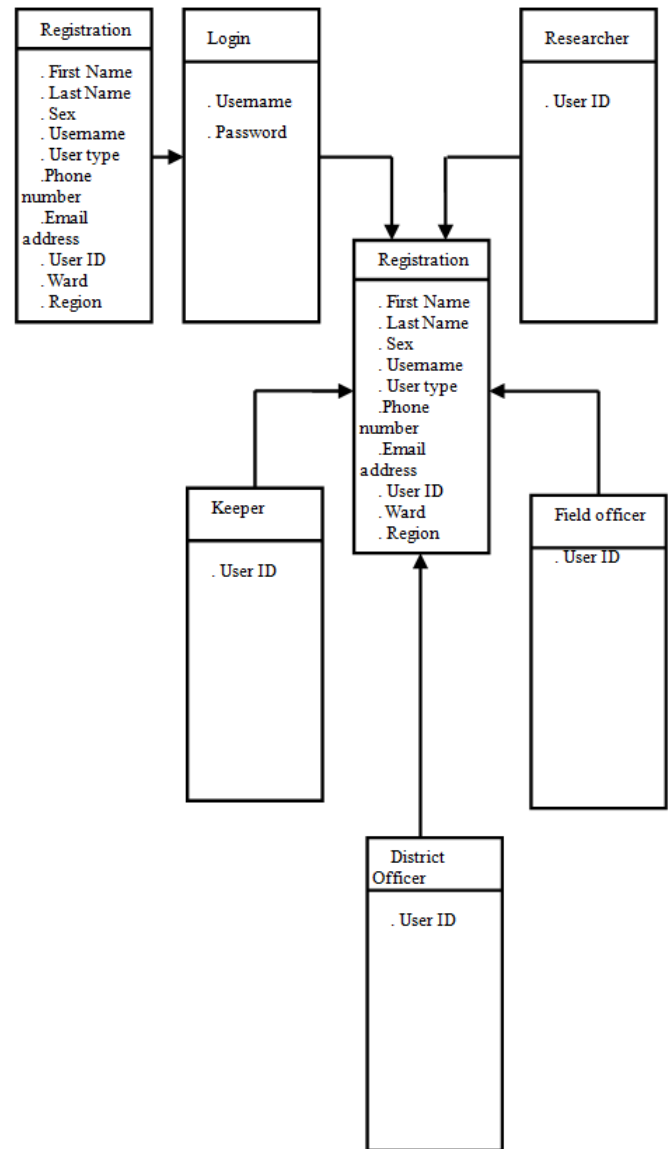


Figure 7. System class diagram.

3.9. Application Wireframe

Wireframes is an important design tool used in mobile app development. It is a visualization tool for presenting proposed functions, structure and content of the mobile app. It separates the graphic elements of the app from the functional elements in such a way that Android development teams can easily explain how users will interact with the app.

A key aspect of mobile applications development is building a good wireframe. Namely, we miss out on the layout configuration at the beginning, our development, thus the project costs could exceed the planned amounts. Nevertheless, changes in the specification are possible at any time. Here, some will change the mobile application price, whereas others will just re-use already allocated resources [16].



Figure 8. Application wireframe.

4. System Overview

In this section, we describe the design of our machine learning - based framework for enhancing performance of livestock mobile application system. First, we introduce the hardware components employed in our system. Next we discuss the software components of the system. Finally, we describe the overall architecture.

4.1. Hardware Components

Hardware comprises of all physical components that make up the intelligent livestock information system.

4.1.1. Smartphone

Smartphone are used to send queries and receive the requests. The user can request the modern livestock practices at any time from the server. The smartphone also has SQLite database to synchronize and store data.

4.1.2. Server

A server is a computer program that provides services to other computer programs (and their users) in the same or other computers. The computer that a server program runs in is also frequently referred to as a server (although it may be used for other purposes as well) [17].

4.1.3. Remote Database

A database is any collection of related data. The database stores details for all the users subscribed in the service with their locations. Hence the proposed system will use remote database to store data. The data replication will be done using

SQLite.

4.2. Software Components

These are the instructions that operate the functions of the information system.

4.2.1. PHP

PHP is a server-side scripting language designed for web development but also used as a general-purpose programming language [18]. It provides a powerful tool for building dynamic and interactive web page content. PHP is a widely-used, free, and provides an efficient alternative to competitors such as Microsoft's API [18].

4.2.2. Eclipse

Eclipse is an integrated development environment (IDE) which contains a base workspace and an extensible plug-in system for customizing the environment. It is written mostly in Java and can be used to develop applications in other programming languages as well [19]. Eclipse is used to design the Android system. It has XML side, which is used for designing the application, and Java side which is used for coding.

4.2.3. Android Software Development Kit (SDK)

We will use the Android Software Development Kit (SDK) to develop Applications in the Java programming language.

4.2.4. Android Application

Android is the world's most popular operating system for mobile devices and tablets. It is an open source operating system created by Google and is available to all kinds of developers with various expertise levels, ranging from rookie to professional [20].

4.2.5. SQLite

SQLite is the embedded database used for Android applications, which is a client-server relational database management system.

4.2.6. Matlab

We will use MATLAB as it has a number of built-in features that make machine learning easier. MATLAB is also very fast compared to other languages such as Mathematica and Python.

4.3. System Architecture

The ILIS framework will collect livestock data from keepers via mobile phones. The data will be sent to the database either through the regular Internet or via synchronization (in case of poor connectivity). Livestock experts will view data and respond to any query from the keepers. The system will be able to learn and predict the responses using machine learning techniques.

The application architecture and its working principle are centered on the use of machine learning techniques.

The centralized database provides a livestock database to the users. The purpose of choosing centralized data sharing is

to reduce the storage overhead on each phone. Otherwise, each mobile may have its own local database stored on the device and sharing may be allowed between different users.

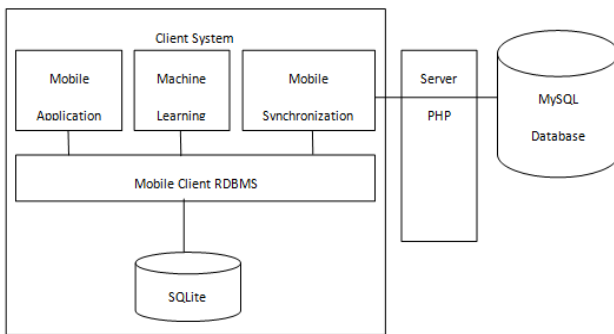


Figure 9. System Architecture.

4.3.1. Client System

The Android mobile application will be designed and installed on a smartphone and will provide an easy-to-use Graphical User Interface (GUI). The application will request resources from the server via the Internet or with a limited connectivity and will check connectivity state using a connection manager.

4.3.2. Typical Learning

Smartphone devices will be installed with a developed android application. The designed system will be able to replicate from the remote database to the SQLite. Machine learning will be able to use experience and make predictions. Server can learn to provide feedback automatically. Server will be trained automatically by mining and analyzing data of all the possible livestock practice scenarios and become a real intelligent one.

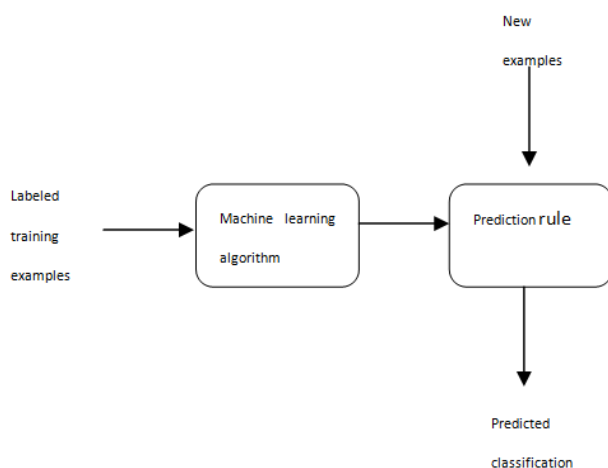


Figure 10. Typical learning.

4.3.3. Steps in Developing Machine Learning Application

Below are steps that will be used in developing machine learning application.

- 1) Collect data. Data will be collected from livestock keepers and livestock experts from fifteen wards of Arumeru district, Arusha region, in Tanzania.

- 2) Prepare the input data. We will prepare the input data in a useable format.
- 3) Analyze the input data. We will look at the data from the previous task to make sure steps 1 and 2 are actually working, and to see if we can recognize any patterns or if there will be anything obvious. We will plot data in graphs.
- 4) Train the algorithm. This is where the machine learning takes place. We will feed the algorithm good clean data from the first two steps and extract knowledge or information.
- 5) Test the algorithm. We will test it to see how well it is doing. We will use some known values to evaluate the algorithm.
- 6) Use it. Here we will make a real program to do some task, and see if all the previous steps work as expected.

5. Conclusion

This paper proposes the design of a machine learning - based livestock information system. It discusses different modeling diagrams, software, hardware descriptions, and architecture of the system. The system is designed to work in poor connectivity rural areas. The requirements of livestock keepers and livestock experts are analyzed to develop the main system functions and system architecture.

This Intelligent Livestock Information System (ILIS) will be able to provide and predict feedback to the livestock keepers. This solution will also collect livestock data from livestock keepers through mobile phones. The data will then be sent to the database if connectivity is available or through synchronization if connectivity is poor. Overall this solution embodies an artificial intelligence approach which combines hardware and software technologies. The design will leverage the Android ADK operating system and Android mobile devices or tablets.

Our main contribution here is the designing of intelligent livestock Information System, which is a novel idea in the field of mobile livestock information systems. It will be able to respond to user queries and make predictions. The System will provide livestock services to anyone at anytime, overcoming the constraints of place, time and character.

The ILIS will minimize human operations and supervisions, so as to reduce the cost of manpower and the lost from human mistake and to enhance the extension services.

6. Future Work

In the future we plan develop and implement machine learning – based application framework for enhancing performance of livestock mobile application systems. The application will cutter for the problem of decreasing performance of mobile apps when in poor connectivity areas. When completed, this system will enable users of an ILIS, in areas of poor connectivity, to experience same performance levels as in good connectivity areas.

Acknowledgement

We hereby express our gratitude to the Nelson Mandela African Institution of Science and Technology (NM-AIST), Livestock Training Agency (LITA-Tengeru), National Artificial Insemination Agency (NAIC), and Arumeru District Council for their support.

We acknowledge the input made by the anonymous peer reviewers.

References

- [1] Weld, D.S., editor. (1995), The Role of Intelligent Systems in the National Information Infrastructure, American Association for Artificial Intelligence.
- [2] Oksana Pomorova, Tetyana Hovorushchenko. Intelligent Assessment and Prediction of Software Characteristics at the Design Stage, American Journal of Software Engineering and Applications. Vol. 2, No. 2, 2013, pp. 25-31. doi: 10.11648/j.ajsea.20130202.11
- [3] Noble, B. D., Satyanarayanan, M. C., Siewiorek, D. P., Zhang, H. and Katz, R. H., 1998. Mobile Data Access, School of Computer Science, Carnegie Mellon University, CMU-CS-98-118.
- [4] Terry, D. B., Theimer, M. M., Petersen, K. and Demers, A. J., 1995. A Portable Multimedia, Terminal Weakly Connected Replicated Storage System, In Proceedings of the 15th ACM Symposium on Operating System Principles, 30(12), 64–76. PMID:10152743
- [5] Applion, “cow tracker”, accessed on 25th March 2015, from <http://applion.jp/android/app/com.sbumpkin/>
- [6] Freapp, “Mobile cattle tracker”, accessed on 25th March 2015, from <http://app.freapp.com/apps/ios/775966992/>
- [7] Google play, “Livestock feeding made easy”, accessed on 23rd March 2015, from <https://play.google.com/store/apps/details?id=com.livestockfeeding>
- [8] DARE/88 ARE/ICAR ANNUAL REPORT 2003–2004, “Livestock Information Management System”, unpublished.
- [9] Metz, T., Asfaw (1999), Livestock Information Management System, ILRI, Ethiopia.
- [10] Joris Vanderschrick (2011), System Requirements Analysis: The first step to value-based system development, Uit: Handboek Requirements.
- [11] Marshall B. Romney and Paul John Steinbart (2003), Accounting Information Systems, 9th Edition, Prentice Hall Business Publishing.
- [12] IBM, “UML basics: An introduction to the Unified Modeling Language”, accessed on 15th March 2015, from <http://www.ibm.com/developerworks/rational/library/769.html>
- [13] Steve Hoberman (2009), Data Modeling Made Simple 2nd Edition, Technics Publications, LLC 2009.
- [14] Steve Hoberman (2014). Data Modeling Master Class Training Manual 5th Edition. Technics Publications, LLC.
- [15] Webopodia, “entity-relationship diagram (model)”, accessed on 10th March 2015 from http://www.webopedia.com/TERM/E/entity_relationship_diagram.html
- [16] Profi.co, Mobile Applications Development. Accessed on 20th March, 2015 on <http://profi.co/mobile-applications-development/>
- [17] WhatIs, “Server”, accessed on 19th March 2015, from <http://whatIs.techtarget.com/definition/server>
- [18] W3Schools, “Introduction to PHP”, accessed on 20th March from <http://www.w3schools.in/php/intro/>.
- [19] Wikipedia, “Eclipse (software)”, accessed on 20th March 2015, from http://en.wikipedia.org/wiki/Eclipse_%28software%29.
- [20] Cprogramming, “Developing for Android - An Introduction”, accessed on 21st March 2015, from http://www.cprogramming.com/android/android_getting_start ed.html.