HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Master's thesis

Geography

Geoinformatics

SPATIOTEMPORAL MODELLING OF ROOFTOP RAINWATER HARVESTING WITH LiDAR DATA IN THE TAITA HILLS, KENYA

Oyelowo Oyedayo

2018

Supervisors:

Petri Pellikka

Mika Siljander

UNIVERSITY OF HELSINKI

DEPARTMENT OF GEOSCIENCES AND GEOGRAPHY

DIVISION OF GEOGRAPHY

P.O. 64 (Gustaf Hällströmin katu 2)

FI-00014 University of Helsinki Finland

Tiedekunta/Osasto Fakultet/Sektion – Faculty	Laitos/Institution– Department
Faculty of Science	Department of Geosciences and Geography

Tekijä/Författare – Author

Oyelowo Oyedayo

Työn nimi / Arbetets titel – Title

Spatiotemporal Modelling of Rooftop Rainwater Harvesting with LiDAR Data in the Taita Hills, Kenya

Oppiaine /Läroämne – Subject

Geoinformatics

Työn laji/Arbetets art – Level	Aika/Datum – Month and year	Sivumäärä/ Sidoantal – Number of pages
Master's thesis	12/2018	121

Tiivistelmä/Referat – Abstract

The puzzling thing about water is that, while it is very abundant in our planet – earth, millions of people globally face water scarcity. Some places, however, do not. In other places, it is not really that there is no water at all, but it is not available all year round, in most cases. This underscores the importance of putting into cognizance, the spatial and temporal context of water scarcity, hence, the basis for this project.

Developing countries, especially have had worse situations with water scarcity due to population explosion and the lack of the technological advancement to harness, purify, transport, store, deliver and reuse water. One of such countries is Kenya, where many do not have access to potable water. Many solutions have been proffered without adequately addressing the issue itself. Rooftop rainwater harvesting is a potential solution to ameliorate this problem.

In this thesis, I took a holistic approach to evaluate the potential of Rooftop Rainwater Harvesting (RRWH) in meeting the domestic water needs of the Taita People, Kenya. Importantly, contrary to other RRWH studies, I attempt to introduce and synergize the temporal aspect with the spatial context, in order to deeply understand the monthly dynamics of RRWH. This is crucial in answering the 'where' and 'when' questions of RRWH. This aims to provide a decision support for stakeholders, by presenting the results visually and quantifiably.

The project is mainly divided into three parts. The first part involves the validation and utilization of a Light and Range Detection (LiDAR) data, for automatically generating the footprints of roofs in Taita. Herein, I compared the accuracies of LiDAR datasets from same area but different years. The second part utilizes the roofs' polygons generated from the LiDAR data to estimate the Rooftop Rainwater Harvesting Potential in the region, by integrating it with Climatologies at high resolution for the earth's land surface areas (CHELSA) and a strategically chosen universal roof coefficient. Lastly, household survey was carried out in the study area to understand the social context and integrate the data into my model.

The result shows that there is a clear temporal trend to RRWHP in the area, and a single annual RRWHP model might be too generalized to give sufficient insight into understanding how much the system can mitigate water problem in the area. It also logically incorporates the survey data into the model to provide information about measurable monthly and annual values, as to percentage of the households that RRWH can fulfill their needs.

Avainsanat – Nyckelord – Keywords

Rainwater harvesting, RRWHS, LiDAR, building detection

Säilytyspaikka – Förvaringställe – Where deposited

University of Helsinki, Kumpula Science Library

DEDICATION

To my late parents. Your then 5-year-old boy will make you proud someday.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

AMCOW	African Ministerial Council on Water
DRWH	Domestic Rainwater Harvesting
CIA	Central Intelligence Agency
DEM	Digital Elevation Model
DSM	Digital Surface Model
ECMWF	European Centre for Medium-Range Weather Forecasts
ERA	ECMWF re-analysis
GHCN-M	Global Historical Climatology Network-Monthly
GCOS	Global Climate Observing System
GIS	Geographic Information System
GPCC	Global Precipitation Climatology Centre
ICRAF	International Centre for Research in Agroforestry / World Agroforestry Centre
IDE	Integrated Development Environment
KNBS	Kenya National Bureau of Statistics
LiDAR	Light Detection and Ranging
NDVI	Normalized Differential Vegetation Index
QGIS	Quantum Geographical Information System
RRWH	Rooftop Rainwater Harvesting
RRWHS	Rooftop Rainwater Harvesting System
RRWHP	Rooftop Rainwater Harvesting Potential
SDG	Sustainable Development Goals
SPSS	Statistical Package for the Social Sciences
UNEP	United Nations Environmental Programme
WCRP	World Climate Research Programme
WGS 84	World Geodetic Survey 84
WSSD	World Summit for Sustainable Development
STD	Standard Deviation

1 INTRODUCTION

1.1 Spatiotemporal water scarcity: assessing rainwater harvesting potential (RRWHP) in a holistic way

"There is no absolute water scarcity, there is spatiotemporal scarcity – Oyelowo 2018"

Water is one of the most crucial resources, necessary for the survival and sustenance of our society. Despite this, many people lack access to potable and regular water supply. In 2015, 29 % of World's population was lacking the access, which is a major setback to socioeconomic development (SDG6, 2018). Water and sanitation have been linked to the alleviation of poverty and realization of sustainable development (WSSD, 2002).

Africa has about 3,991km^3 annual freshwater resources, but many people continue to face water scarcity. Water poverty in Africa has stemmed also from the 40 billion hours being wasted on carrying water mostly by females, while the effective solution would be to have close access to water from home. This is especially important in areas with sparse population where it might be prohibitively high to harness surface and groundwater resources. Rainwater is far from being maximally utilized in Africa (ICRAF & UNEP, 2005).

In recent years, there has been an unprecedented increase in rainwater harvesting in Kenya, in order to answer to water problems prevalent all around the country (Recha, Mukopi, & Otieno, 2015). Water harvesting has proved to be very useful in mitigating water scarcity during the dry seasons as it provides water for domestic, agricultural and even commercial purposes (Gould, 2015; J. Mwenge Kahinda, Taigbenu, & Boroto, 2010). By extension, this provides means to ameliorate poverty, as water plays a critical role in health and food security of the people. Hence, it is crucial to secure and manage this important resource in every community (Priscoli, 2004).

Furthermore, rain water harvesting has been recognized by the government, NGOs and international bodies as a viable means to mitigate the water problems plaguing the country in a decentralized manner. Despite the known benefits, quite several projects degrade in a short period

and some do not even see the light of the day. Thus, this raises the question of sustainability which must be addressed, in order to provide a long-term solution to water problems, especially in the rural communities (Thomas & Martinson, 2007).

This study investigates the status quo, major socio-economic challenges, and the needs and the viability of rooftop rainwater harvesting in addressing these issues sustainably and scalable manner in the Taita Hills, Kenya.

1.2 Rationale behind study: Spatiotemporal water scarcity

Over the years, stakeholders and organizations have had slow progress in the advocacy of rainwater harvesting. A major issue they have faced has been the inability to find concrete and substantiated scientific findings to show and convince policy makers where rainwater harvesting can be effectually utilized. Such need has necessitated the use of GIS to manipulate, manage and utilize data to disseminate information in a simple way (ICRAF & UNEP, 2005).

While many studies have assessed rainwater harvesting, not so much has been done specifically on rooftop rainwater harvesting. Only a few studies have assessed rooftop rainwater harvesting potential (RRWHP), but they have only given a broad generalization by focusing on the spatial aspect and totally neglecting the temporal aspect of it (Gaikwad, 2015; Giridhar, S, & Viswanadh, 2013; Shadeed & Lange, 2010). Studies have not been able to provide a synergistic framework for accessing RRWHP spatiotemporally (Dadhich & Mathur, 2016; Giridhar et al., 2013; Liaw & Chiang, 2014; Ojwang, Dietrich, Anebagilu, Beyer, & Rottensteiner, 2017; Traboulsi & Traboulsi, 2017; Zain M. Al-Houri, Oday K. Abu-Hadba, & Khaled A. Hamdan, 2014; Zende, 2015). My opinion is that water is not necessarily absolutely scarce, but spatiotemporally scarce. Therefore, I refer it to be “spatiotemporal water scarcity”. In most scenarios, water scarcity is seasonal across space. This underscores the importance of considering the temporal aspect alongside the spatial context, to give a deeper insight into how the RWHS can be better utilized and supplemented when temporally scarce.

Recent studies in rainwater harvesting cover only small areas due to the manual digitization of roofs to estimate RRWHP (Dadhich & Mathur, 2016). It is very cumbersome and requires much

manpower to digitize many buildings. This is inefficient when covering large areas as humans are prone to errors, as well as inaccurate as buildings can also be omitted. On the other hand, other studies which have utilized automated building extraction do not go further in using the result for water harvesting application (El-Deen Taha & Ibrahim, 2016; Hu & Ye, 2013; Wang, Lodha, & Helmbold, 2007). Furthermore, the extraction methods are usually not very open and repeatable; and/or substantially validated (Lupia, Baiocchi, Lelo, & Pulighe, 2017; Nthuni, Lübker, & Schaab, 2014; Shinde & Gaikwad, 2016). Therefore, I have adopted an automatic approach in the extraction of footprints of buildings' roofs in Taita Hills from LiDAR data, with measurable and open validation which compares LiDAR data from two different airborne campaigns in the same area. The two data have different quality in terms of point density. I have evaluated the effect of the quality of the data on the accuracy of feature extraction, which is how well to extract buildings. A rainfall model generated from CHELSA to estimate precipitation for each month in the Taita Hills was validated with precipitation data from the ground weather stations managed by Taita Research Station of the University of Helsinki.

Lastly, considering that water scarcity is a problem that affect the people, it is salient to also understand the social context by having a physical presence and firsthand feel of the water related issues in the locality. With these, water problems can be evaluated holistically and logically connected to the derived RRWHP model from geospatial analysis.

1.3 Objectives of this thesis

1.3.1 Main objectives

a) To provide a novel framework for understanding the spatiotemporal pattern of rooftop rainwater harvesting potential in Taita Hills region, and thus, create a decision support for RRWH implementation in the Taita Hills area.
b) To understand the spatiotemporal pattern of potential satisfaction of the people's water-use by rooftop rainwater harvesting.

1.3.2 Other objectives

a) Automate the process of creating, regularizing and validating buildings geodatabase for Taita Hills.

b) Understand the effect of lidar's pulse density and point spacing on the accuracy of buildings' footprints extraction.
c) Validate rainfall model for the area.
d) Understand the social context of water related problems in the area

1.4 Research questions

c) Is there a distinct spatiotemporal pattern to rooftop rainwater harvesting in Taita? Is water absolutely scarce or spatiotemporally scarce?
d) How can the spatiotemporal modelling be well combined with the social context, to give a holistic picture about rooftop rainwater harvesting potential?
e) Does the water consumption vary between the highlands and the lowlands of Taita Hills?
f) What challenges do people face in getting water? Do they vary considerably between the highland and lowland?
g) Is the rooftop rainwater harvesting system a viable solution?
h) How can the LiDAR data and rainfall data be validated?

1.5 General methodology and structure of the thesis

This project is divided into seven chapters. In the first chapter, I give a general overview of present water situation in the region and my approach to understand and investigate the problem to offer a sustainable solution. The second chapter dives deeper into what the system is like and the approaches of previous studies. Chapter three describes the physical and socioeconomic aspects of the study area. Thereafter, I describe my data in chapter four. The methodologies and results are presented in chapters five and six, respectively. Lastly, I end by recapitulating, evaluating and concluding the study in chapters seven and eight. Figure 1 gives a general idea of how the entire work flow of this project was executed.

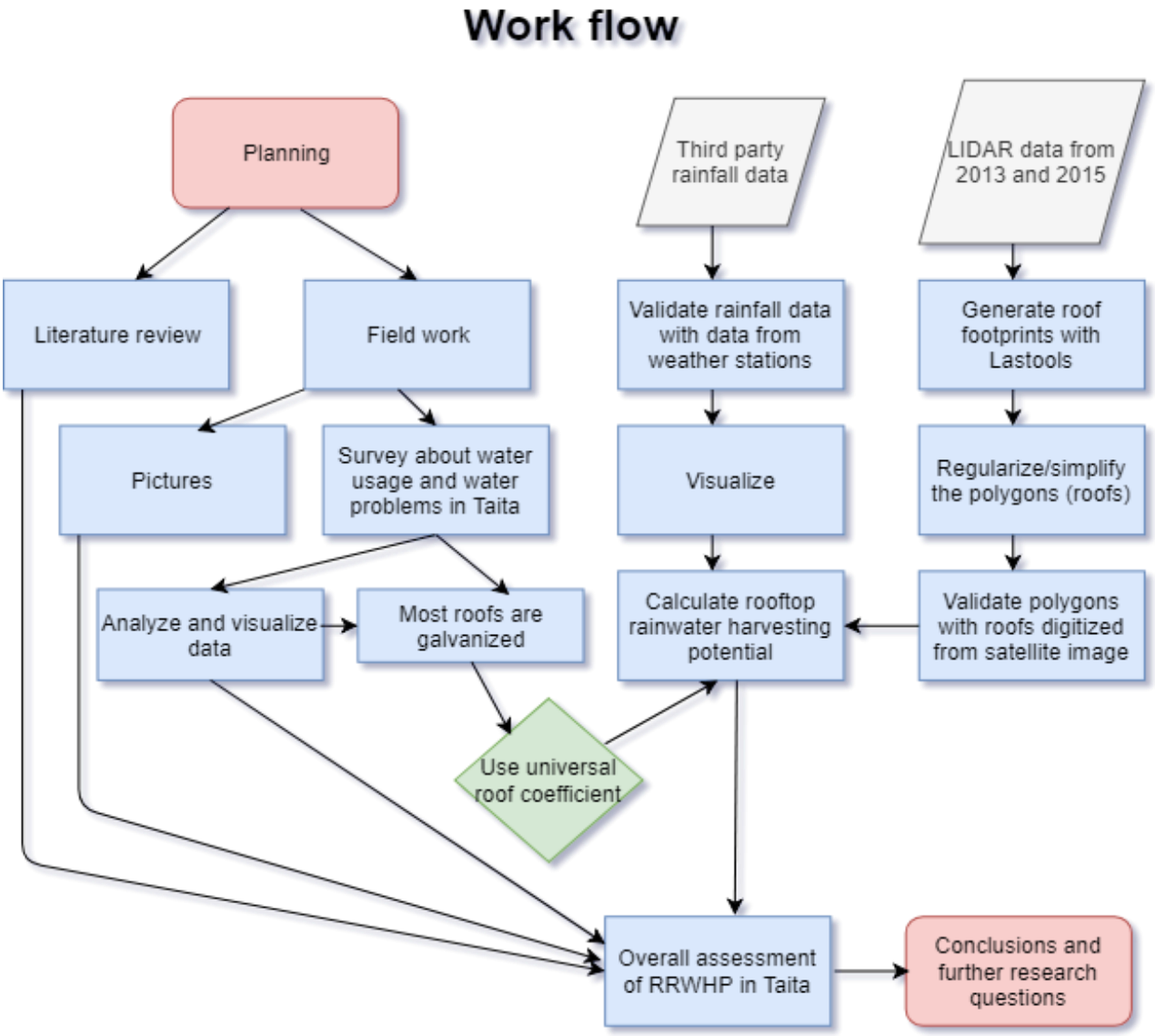


Figure 1. The workflow of the project.

2 BACKGROUND

2.1 Rainwater harvesting

Rainwater harvesting is the collection and storage of rainwater runoff for use (Siegert, Chapman, & Finkel, 1991). Therefore, rainfall can be used more efficiently for domestic and non-domestic purposes by capturing it onsite where it falls (Ziadat, Mazahreh, & Oweis, 2006). This originates from over 4 millennia ago (Critchley, 1989). This water can be used for domestic purposes to supplement the already existing sources or for biomass production (Malesu & Resilience, 2014; Mati et al., 2007). Rainwater harvesting systems have been classified based on the used catchment surface such as roof catchment, groundwater catchment systems, dams and rock catchment (Gould, 1999, 2015). This paper focuses on the roof catchment system.

RRWH technique involves catching rainwater from the rooftop and channeling it through gutters, which can then be stored in tanks and utilized thereafter. It provides an inexpensive, convenient accessible water source for domestic purposes. The water is delivered directly to the house, hence, not affected by topography or geology and can be managed at household level (Figure 2). It also poses lower chemical and biological risk. Furthermore, roof rainwater harvesting can also help to reduce pressure on all these other water sources. This is especially important in addressing the increasing water scarcity and mitigating the consequences of climate change in the region.

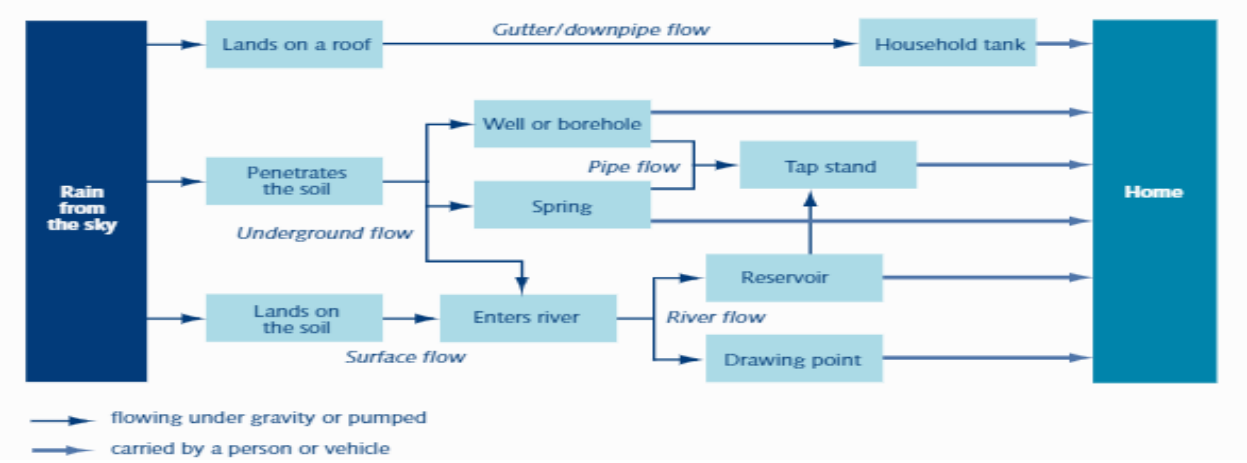


Figure 2. Rainwater path (Thomas & Martinson, 2007).

Rooftop rainwater harvesting can help to provide water to thousands of households, thus, it is encouraged to have water tank as much as possible in every household. Compared to other means, the existing roof structure poses a lesser negative effect on the environment for the development of water resources (Shadeed & Lange, 2010). Interestingly, rainwater collected from the roof is less contaminated than the surface runoff (Thomas & Martinson, 2007). The RRWHS can also help the people to reduce the dependency of the people on the government on water supply, as this can be unreliable. In Figure 3, according to an anecdotal information by a respondent these pipes are meant for a water project in Taita Hills but have been left unused for two years.

Figure 3. Abandoned water project in the Taita Hills (Oyelowo, 2018).

2.2 Description of rainwater harvesting system

The RRWHS comprises three major parts. This includes the catchment (roof), the gutter (runoff delivery system) and a storage tank (Figure 4).

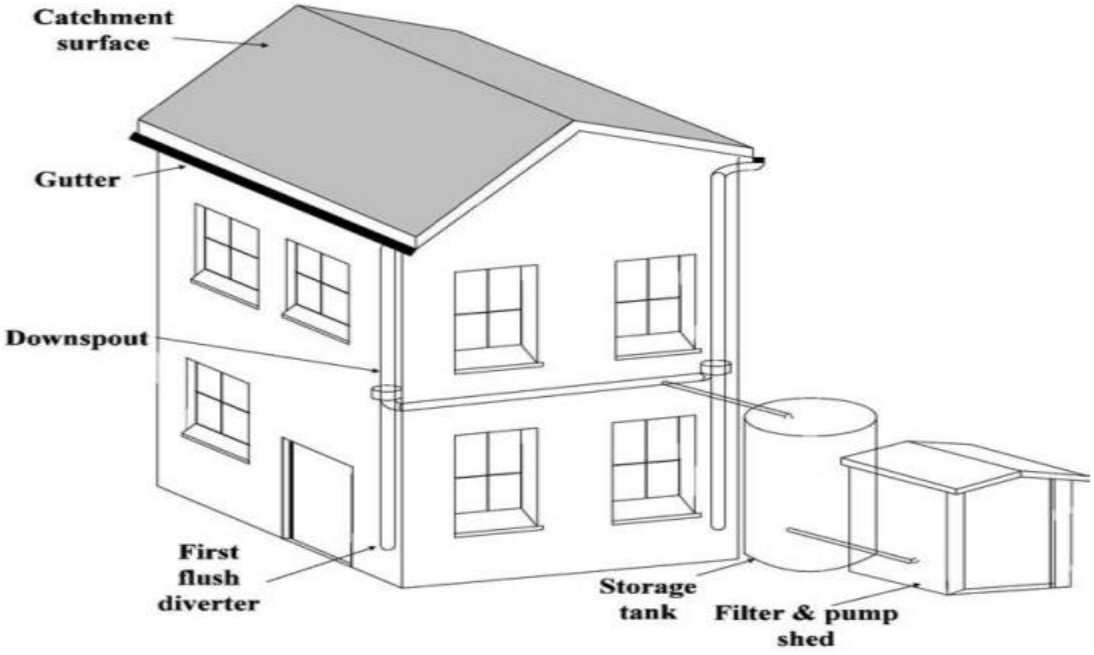


Figure 4. Parts of rooftop rainwater harvesting system (Zhe Li & Fergal Boyle, 2010).

2.2.1 Catchment

Typically, the catchment is the rooftop for domestic rainwater harvesting. Hence, it should be impermeable and safe for the rainwater. The effectiveness of the system and the water quality mainly depends on the roof material and the effective roof area (Abdulla, 2009). Such effective roof materials include the galvanized, tiles and corrugated-iron sheets (Zhe Li & Fergal Boyle, 2010). The preferred qualities of these materials are the following: smoothness, impermeability, accessibility, affordability. Clean flat cement roof can also be utilized (Gould, 2015).

By and large, the surface of the roof should be unpainted and uncoated. Additionally, an effective roof catchment is smooth, considerably large enough to provide good water quality and enough water quantity. An efficient roof usually has high runoff coefficient ranging from 70 % to 80 %. The quantity of the water delivered to the storage tank is reduced by spillage, leakage and evaporation (Zhe Li & Fergal Boyle, 2010).

2.2.2 Gutter

This includes the gutters and pipes that transport the roof rainwater into the storage tank. These gutters are usually around the edges of the roof hanging close to the eaves, with the downpipes receiving the water into the tank. It is recommended to have a minimum of 1cm^2 cross-sectional area of gutter per m^2 of roof area (Gould, 2015). This is crucial to capture considerable amount of water and with support of a splash guard, can prevent overflow. Furthermore, filters and/or other cleaning materials are used to prevent the entry of foreign materials such as leaves, stones, insects etc. (Zhe Li & Fergal Boyle, 2010).

The downpipes could have a smaller cross-sectional area as it is usually positioned vertically and can deliver water faster than it receives. It is also important to do a regular maintenance to keep the system effectual. Mainly, a well-designed and effectual system could deliver 90 % of the water into the storage tank. However, this could be also lesser to about 70-80 %.

2.2.3 The Storage Tank

This is the container where the rainwater is delivered into and stored. Generally, the tank is the most expensive part of the system, which could take more than half of the entire cost. Therefore, the tank must be made properly. It is easy to detect leakages, damages and drain out dirty water in above-ground tanks, thus cheaper maintenance cost. Also, water can more easily be delivered for domestic use via gravity, without having to pump it up like the underground counterpart. In addition, underground tanks are much more expensive to construct due to the excavation cost. They are also susceptible to ground and flood water contamination. Nonetheless, aboveground tanks also take up space and more exposed to the hot temperature. All in all, the advantages of the aboveground outweighs the disadvantages, when compared to the underground tanks (Traboulsi & Traboulsi, 2017).

Furthermore, tanks could be rectangular, square or circular in shape. However, the rectangular and square are recommended for above-ground location because they are cheaper while the circular tanks for underground to withstand pressure from soil exerted on the wall of the tank when empty. Common materials used include the concrete cement, plastic, clay and metal.

Another thing to bear in mind is the size of the tank. Storage tanks should be appropriately sized. An undersized tank might not provide enough water for the household as the tank is filled very fast and exhausted quickly, too. On the other hand, an oversized tank will not only be unnecessarily too expensive but also have infrequent cycling. Consequently, the water quality diminishes as the water might hardly be exhausted for fresh rainwater to come in. This will also depend on the household size and other uses such as farming (Zhe Li, Fergal Boyle, 2010).

The construction of RWH tanks in Kenya dates to early 1900's. However, they became more common during recent decades. The corrugated galvanized iron sheets were in vogue in the 1970's and 1980's, but due to corrosion, these tanks were dumped in no time. Some NGOs also promoted another kind of tanks made of stick and branches, but they were quickly destroyed by insects such as termites. In 1980, the ferro-cement tank first constructed in Kibwezi, Kenya in 1978 became widespread. However, due to the substandard materials used by the builders, these tanks had several issues, but afterwards solving the issues they have become successful and ubiquitous in Kenya due to their affordability and durability. Plastic tanks made by local companies have also become popular (Ojwang et al., 2017). Figure 5 shows various kinds of water storage tanks used in Taita Hills.

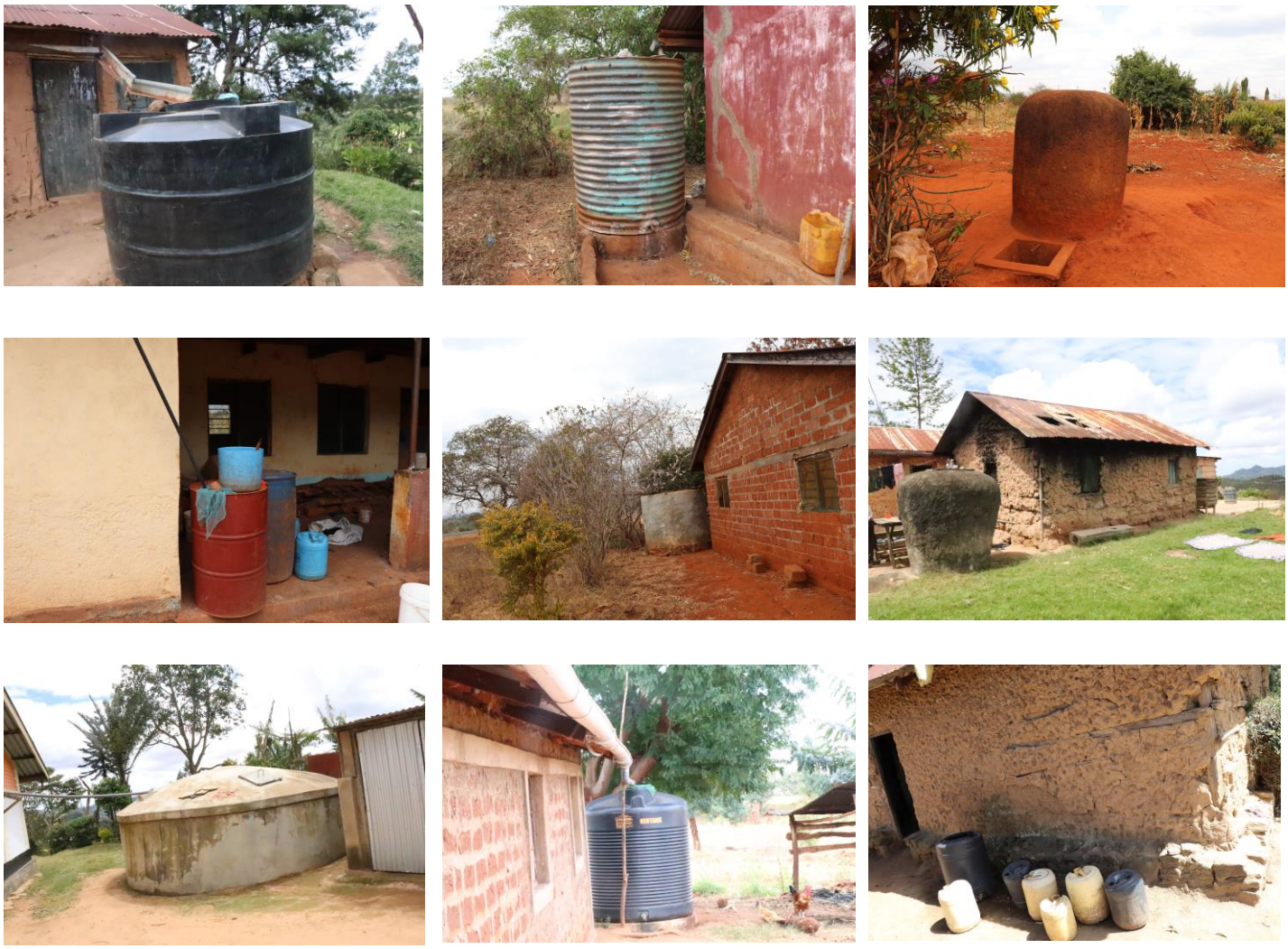


Figure 5. Various types of storage tanks used in Taita Hills.

2.3 Domestically Harvested Rainwater

2.3.1 Quality

The quality of water derived from roof rainwater harvesting depends on the design, maintenance, materials used, roof cleanliness and environment. Dust, leaves, insects, chemical deposits and birds' droppings are major sources of contamination for rooftop RWH tanks. If properly designed and maintained, they can provide a good quality water (Zhe Li & Fergal Boyle, 2010).

The roof is the most contaminated part of the system as it is very exposed. It contains particles, heavy metals and microorganism, for example. As a result, the first few millimeters of rainwater should be diverted away from the tank, in the other words to wash the roof. This is referred to as the first flush, after which the rainwater is harvested (J. Mwenge Kahinda, Taigbenu, Sejamoholo, Lillie, & Boroto, 2009). A technique has also been created to determine the amount of first flush (Thomas & Martinson, 2007). It is recommended to have the storage tanks cleaned regularly, to remove particles and substances that might be in the tank, if the water is rarely exhausted. Also, the tank should be properly covered to prevent insects leaves and other foreign materials (Zhe Li & Fergal Boyle, 2010).

Birds and other animals could infect the roof and tanks with bacteria, protozoa and other micro-organisms (Jean Marc Mwenge Kahinda, Taigbenu, & Boroto, 2007), which renders the water not fit for drinking. In the absence of light and organic material, bacteria and pathogens die slowly after some days (Gould, 2015).

2.3.2 Water-Related Diseases

Domestic rainwater harvesting provides a convenient means of getting access to water without having to walk long distance, especially in developing countries (Jean Marc Mwenge Kahinda et al., 2007)**.** The hygiene of a household and public health is dependent on the amount of water that can be accessible (Bartram & Howard, 2003).

Water is an essential integral part of life and lack of it aggravates the challenges of susceptible groups including those infected by diseases, children, elderly, handicapped and those exposed to waterborne diseases. This is a major impedance to escaping poverty by these groups. Many households only have access to poor quality water. Domestic Rainwater Harvesting (DRWH) can provide much cleaner water for these households. There is insufficient data on the water quality of rainwater in quality. This data inadequacy in a global phenomenon and worse in developing countries (Jean Marc Mwenge Kahinda et al., 2007).

There are dichotomous conclusions in various literature for the quality of DRWH. Some studies show that the roof water quality is potable enough, and is in line with the international guidelines (Ojwang et al., 2017). Conversely, some studies have reported that the physicochemical and microbiological contamination do not meet up with these international guidelines of potable water (Abbott et al., 2006; Nevondo and Cloete, 1999). Vector insects such as mosquito, also can use the water in the storage tank and this is a source of malaria. **Error! Reference source not found.** shows the path of contamination of the water (Jean Marc Mwenge Kahinda et al., 2007).

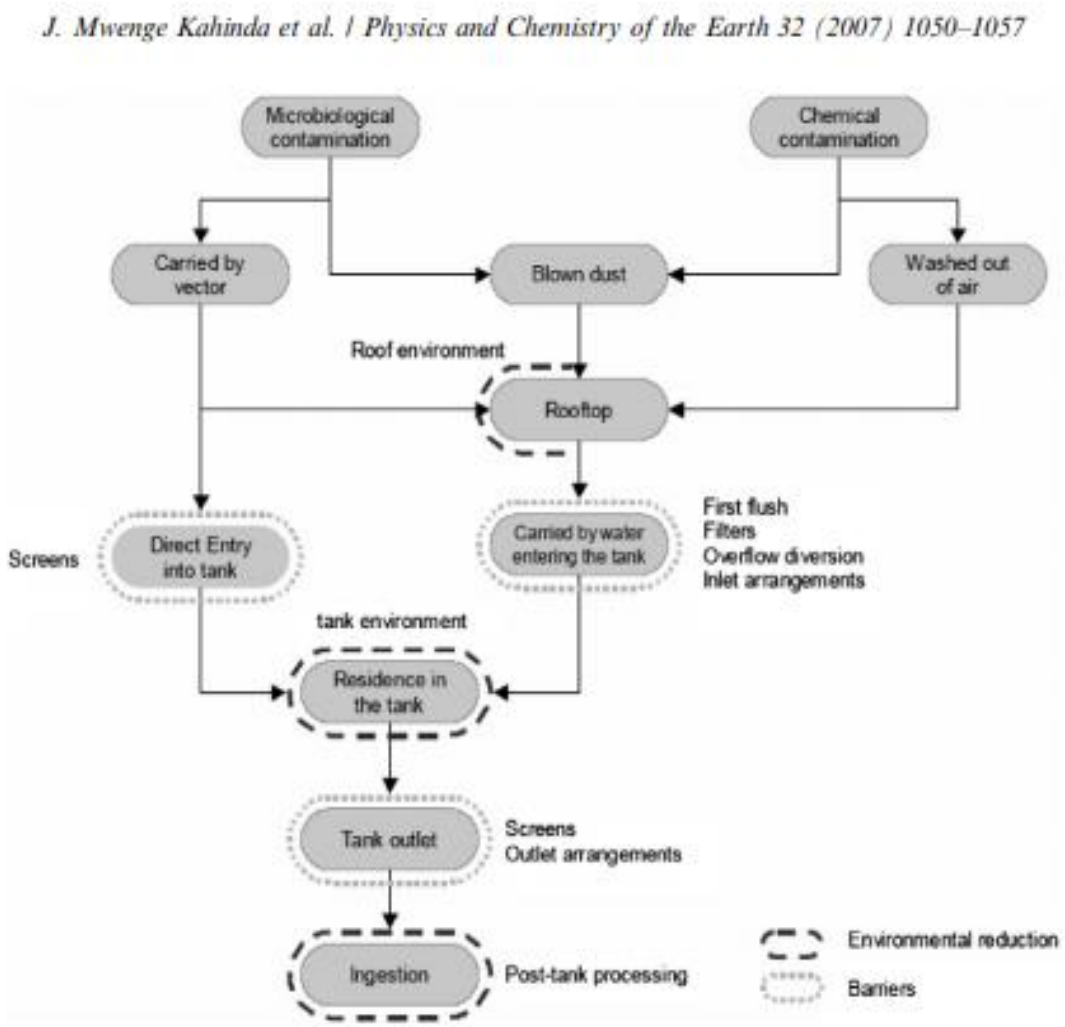


Figure 6. Contamination paths DRWH systems collecting water from rooftop (Kahinda et al., 2007).

2.3.3 Treatment

Due to the water related diseases, it is very critical to treat the rainwater. A simple treatment could increase the water quality significantly for other domestic purposes, besides drinking. However, it requires more expensive and advanced techniques to make the quality of sufficient drinking quality. Simple treatments include chlorination, disinfection, slow sand filtration, and pasteurization (Zhe Li & Fergal Boyle, 2010).

Chlorination is the mostly used and affordable means to easily exterminate most microorganisms. This is used after the water is transported from the storage tank as it might react with some organic materials to produce some unpleasant substances that settle on the tank's bottom. 0.4–0.5 mg/l free chlorine is the recommended and effective level of chlorination (O.R. Al-Jayyousi, 2003). However, some parasites have been shown to withstand low level of chlorines. Disinfection can also be applied to the rainwater to reduce microbiological contamination (Sazakli, 2007). Slow sand filtration, on the other hand, uses sands to filter the water, with the coarsest sand particles coming before the finest at the bottom. This filtration process is more of a biological treatment than physical (Helmreich & Horn, 2009). The efficacy is depended on a constant flow of water through it. This technique is only able to decrease, rather than exterminate the microorganisms in the water (Zhe Li & Fergal Boyle, 2010).

Lastly, pasteurization is a reliable and inexpensive water treatment technique utilizing the heat energy from solar radiation. Temperature of above 50°C is crucial for the effectiveness of this technique and with complete oxygenation of the water. This technique is effective for treating some bacteria, but is limited beyond 10mg/l suspended solids concentration (Helmreich & Horn, 2009).

From the foregoing, a combination of the treatment techniques would be required for an effective water treatment. It is enough to utilize the simplest method for domestic use like toilet flushing. This water would not be suitable for consumption, nonetheless, using a complex means would be prohibitively expensive. Membrane filtration and disinfection system can be used for more complex and effective treatment for potable water. Boiling also helps to treat against larger

viruses and bacteria before drinking. However, this method calls for a lot of maintenance, hence, much more cost (Zhe Li & Fergal Boyle, 2010).

2.3.4 Benefits

The Domestic roof water harvesting is a low cost means of getting water that can help to reduce pressure on other water sources, especially from the community and government. This proves to be useful when there is an interruption from other sources. Besides this, simple treatment would be sufficient and cost effective if the water is not meant for drinking. It can also be easily installed in new and old homes. From environmental perspective, if the system is installed in considerable number of homes, it can help to reduce surface runoff, consequently upon which flooding, and erosion is reduced (Jean Marc Mwenge Kahinda et al., 2010).

2.4 Application of GIS in the evaluation of rooftop rainwater harvesting

2.4.1 Automatic Building Extraction

Geographic information systems (GIS) in combination with hydrological response models, enables the planning and evaluation of runoff harvesting sites which facilitates a rational decision-making system (de Winnaar et al., 2007). It provides a means to collect, store, analyze, describe, transform and show both spatial and aspatial data for specific uses (Coskun & Musaoglu, 2004). By using GIS, potential runoff harvesting sites can be identified with respect to where the runoff is concentrated and can be properly stored and distributed.

Over the years, professionals have widely used building footprints for several purposes. Application includes, the modeling of pollution, disaster planning, forestry, urban planning and modelling, and building extraction (Goodwin, Coops, Tooke, Christen & Voogt, 2009; Jawak, Panditrao, Luis, & Sada, 2014; Wang et al., 2007). These building footprints can be generated manually or based on architectural blueprints. However, these are usually very tedious and require high modelling expertise. Besides, blueprints are mostly available for modern buildings, as opposed to the old ones. Therefore, it is critical to device means to automate the generation of the

footprints that would be cost effective and faster. This will help to speed up several urban modelling and simulations, which will propel the decision-making processes. In this regard, the airborne LiDAR (Light Detection and Ranging) comes into play. The urban models aims to create a 3D navigable area of the city (Wang et al., 2007).

LiDAR is a relatively cheap and accurate technology which is used for creating dense, detailed and accurate 3D point clouds that are spaced irregularly for extracting topographic features (Jawak et al., 2014). It utilizes a laser scanner mounted on an aircraft. The system's components include a laser range finder, GNSS (Global Navigation Satellite System GNSS), inertial measurement unit (IMU) and a computer. The GNSS provides information about the position of the aircraft, in terms of altitude, latitude and longitude (x, y, z) while the IMU obtains the angular attitude of the aircraft as it is taking measurements as roll, pitch and yaw. The IMU provides very high accuracy of all the three dimensions as the aircraft moves vertically and horizontally in flight. Based on this, the pulses are transmitted towards the Earth's surface from the laser scanner, with the records of the laser beams travel time and the energy reflected by the surface.

The aircraft collects data as set of overlapping strips as it moves along several flight lines over an area. Thereafter, the data can be post-processed to differentiate different structures on the Earth's surface (Frueh & Zakhor, 2004). Five various multiple returns can be recorded by the airborne laser scanners, with the first returns coming from roof tops or trees (Figure 7). Some pulses can also penetrate the tree to be intermediate returns or last returns when coming from the ground (Dowman, 2004).

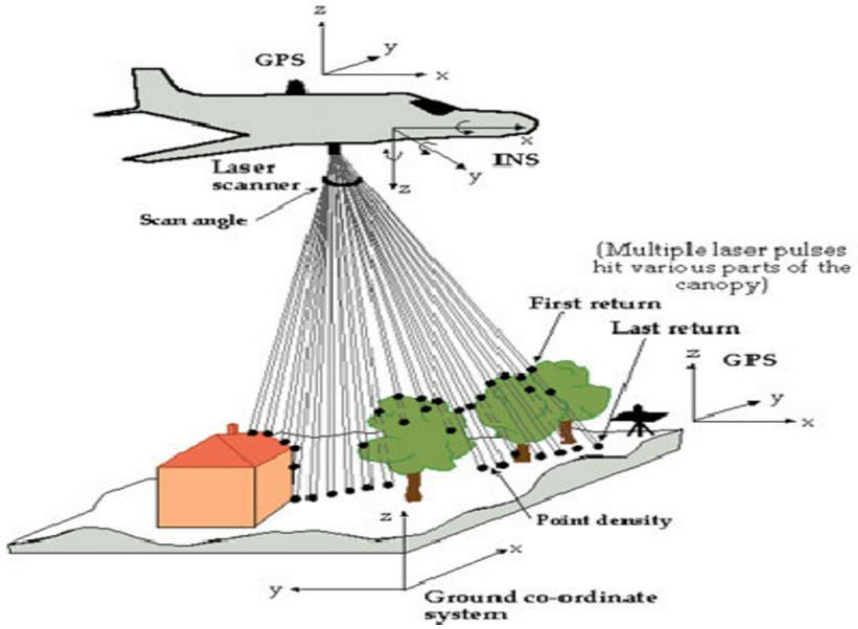


Figure 7: Illustration of LiDAR (Dowman, 2004).

2.4.2 Accuracy Assessment and Validation

The accuracy of LiDAR depends on the sampling density. A major setback in many algorithms that uses data-driven building creation is the presence of a lot of noise in the point around building edges. This noise could be from the sensor, trees or from the result of classification. The accuracy of building footprints generated from LiDAR has been accessed by comparing with those from architectural blueprints or aerial photo. Previous works have used pre-existing building footprints directly for their modelling techniques. This knowledge assumes a priori knowledge of individual locations and shapes of buildings. Multispectral reflectance has also been used to calculate the edges of images (Brenner et al. 2000). Edge detection techniques have also been applied on air photos to determine building boundaries (Rottensteiner et al., 2004). Bayesian networks have also been used for edge detection (Kim & Nevatia, 2004).

Characteristics such as size, shape and height have been used to extract building footprints from DEM or other class (Haithcoat et al., 2001**)**. Thresholds are also applied to remove what could be too small or too big to be a building, for example a car. Thereafter, the rasterized data is used to produce the building footprint and simplified thereafter (Wang et al., 2007). This

simplification assumes some form of orthogonality of the buildings. In order to yield high accuracies, some studies go further to trace and regularize the boundaries of the buildings and filter non-buildings based on the slope or morphological filtering (Sampath et al., 2007; Zhang et al., 2006).

High resolution satellite images have been used for automatic building extraction by utilizing the spectral signatures of houses. However, the lack of information about height makes it difficult to differentiate house from other objects. There has also been attempts to combine spectral data with LiDAR data to utilize the strength of both data types (Ekhtari et al., 2009). LiDAR gives information about height and intensity while high resolution images can give information about spectral signatures, which can help to increase the accuracy of building extraction. LiDAR has a high spatial resolution and is able to detect distances (Lillesand et al., 2008). There has also been an effort to identify and separate buildings in a complex landscape with hills and dense vegetation (Awrangjeb et al., 2012). LiDAR can be used to detect isolated buildings by using identical heights and homogenous Normalized Differential Vegetation Index (NDVI) of the surface. (Awrangjeb et al., 2010).

Various data sources and algorithms are currently used for the extraction of the building footprint. Although many extraction methods that have been devised, most of the evaluation methods have utilized several criteria, which makes it cumbersome to compare the approaches (Wang et al., 2007). They either often use redundant metrics (Awrangjeb et al., 2010) or not empirically evaluate or validate the accuracy of the extraction (Song, 2005). This research tries to use a simple clear-cut technique to evaluate the extraction of the building footprints. The object (building) and the background are considered when doing the evaluation. By comparing with reference objects, a matrix can be created to assess the true positive (TP), true negative (TN), false positive (FP) and false negative (FN) as seen from Table 1.

Table 1. Accuracy metrics (Wang et al., 2007).

TP	Area shared by the extracted and reference objects
TN	Area not in both the extracted and reference objects
FP	Area in the extracted but not the reference object

FN	Area in the reference object but not the extracted object

The above-mentioned metrics are especially common with image classification assessment and give information about the omission and commission errors. However, they have also been adopted in feature-based applications (Awrangjeb et al., 2010). Omission error is the percentage not found in the extracted but in the reference objects while commission error is for the extracted buildings that are detected incorrectly (Song, 2005).

We can determine the completeness or producer's accuracy and the correctness or user's accuracy. Completeness shows the ratio of buildings that are correctly detected to the reference data. The user's accuracy gives information about the buildings that are correctly detected with respect to all the extracted buildings (Wang et al., 2007). Other methods have also been used for measuring shape similarities (Dungan, 2006), but are not well defined (Zeng, Wang, & Lehrbass, 2013). Pixel and object-based analysis have been utilized (Awrangjeb et al., 2010, Rutzinger et al, 2009), with both the raster and vector data models.

Positional accuracy is another important metric, but is rarely used because it is not deemed to be important and also due to the limitations of available data and methods (Zeng et al., 2013). The Root Mean Square Error (RMSE) has also been adopted for assessing the positional errors between the points in the extracted and the reference maps, mostly by using corner points of buildings (Wang et al., 2007). Also, Euclidean distance has been utilized in similar vein, by considering the center of mass of the extracted and reference objects (Song, 2005). In a nutshell, building extraction techniques are assessed mainly based on the shape similarity, matched rate and positional accuracy (Zeng et al., 2013).

2.4.3 Rainfall Modelling

The spatial distribution of rainfall is required for several applications in environmental studies. Such include the hydrologic and ecologic modeling, irrigation scheduling and water

resources management. However, it is more challenging to assess rainfall distribution in mountainous areas due to the topographical effect on the patterns of distribution of rainfall over short distances. A dense network of rain gauges would be required to considerable cover and characterize the rainfall pattern of the area accurately. The cost of installation and maintenance would, however, be prohibitively high. Regardless of the foretasted, hydrologists still have to cover the unrecorded area. To solve this issue, it is important to design the network of the available rain gauge optimally and choose the appropriate interpolation. This necessitates a good understanding of the rainfall spatial variability.

Some of the interpolation techniques that have been adopted include: the station-average, Thiessen polygon, isohyetal methods and inverse distance weighting (IDW) (Thiessen, 1911; McCuen, 1989). In recent times, geostatistical techniques are utilized for estimating rainfall according to regionalize variables (Goovaerts, 1997). Geostatistical methods have been found to derive better precipitation estimates for ungauged locations than conventional methods, as it takes into consideration the spatial correlation between neighboring observations (Campling et al., 2001; Buytaert et al.. 2006). Furthermore, geostatistics allows the incorporation of densely sampled secondary data from satellite and elevation with sparsely sampled primary data from rain gauge.

Kriging is another popular method used in geostatistics for rainfall modelling. It is a generalized least-squares regression method that utilizes the data available in a neighborhood to estimate the values for unsampled areas (Goovaerts, 1997; Deutsch & Journel, 1998). Furthermore, linear regressions can be used by finding the relationship between the rainfall data and the elevation of the area. The elevation is usually generated from a Digital Elevation Model (Daly et al., 1994).

3 STUDY AREA

The study area is located in Taita Taveta County, southern Kenya, which is between 2° 46 south and 4° 10 south and longitude 37° 36 east and 30° 14 east (Taita Taveta County, 2015). Taita Hills is a mountain massif located in a semi-arid part of south-eastern Kenya, approximately 150 km from the Indian Ocean. The highest peak, Vuria reaches up to 2000 meters. Kenya is in the Eastern part of Africa and is bordered by Ethiopia and South Sudan to the north; Somalia to the east; Uganda to the west; and Tanzania to the south. The country covers about 581,309 km^2 with a population of 48 million (CIA, 2013). Figure 8 shows the study areas, which was determined by the extent of the LiDAR data from the 2015 flight campaign.

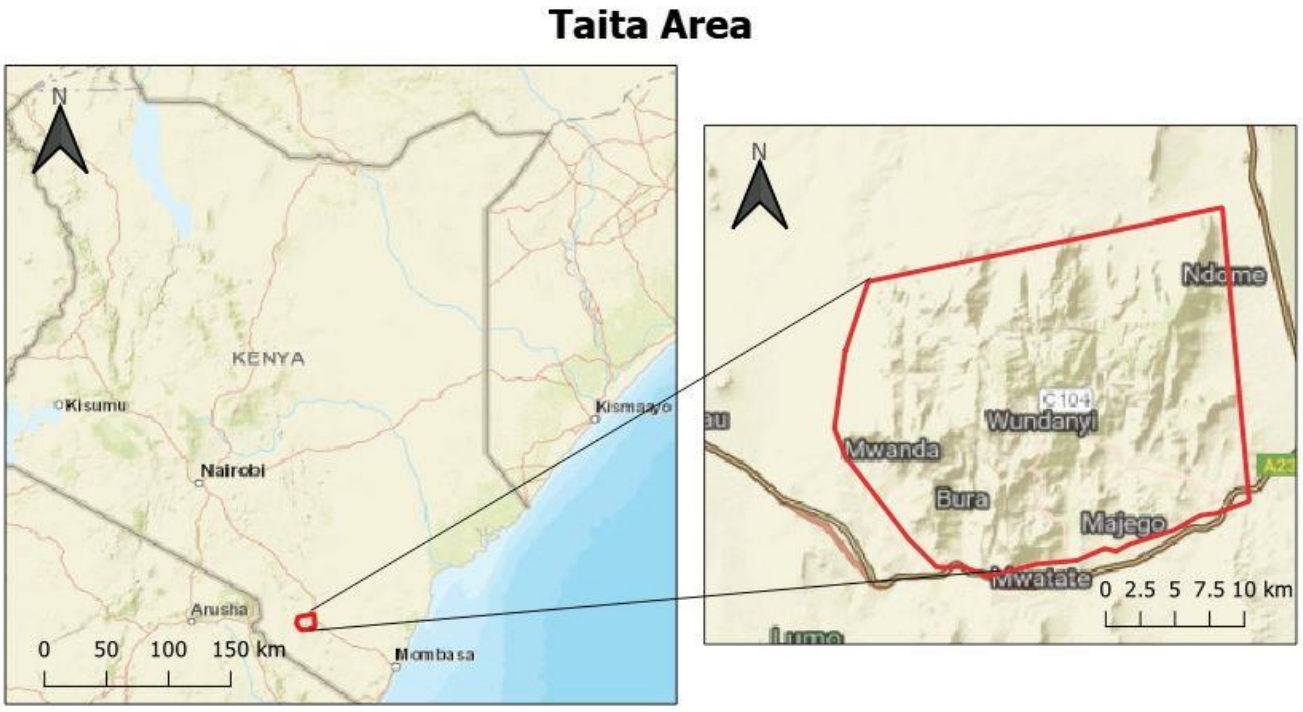


Figure 8. Study area Taita Hills in Taita Taveta County, Kenya.

Taita Taveta County covers about 17,048 km^2, with two national parks included (Taita Taveta County Government, 2013). The Taita Hills of approximately 850 km^2 consists of bedrock from Precambrian era surrounded by Tsavo plains of about 1000 meter above sea level. The rainfall pattern is affected by the movement of the Intertropical Convergence Zone (ITCZ), which causes bimodal rainfall pattern and the orographic effect which causes the uplift of the humid air masses from the Indian Ocean. As a result, there are two rainy seasons in a year, between March and May;

and November to December, respectively. The annual rainfall is as much as 1500 mm (Erdogan et al., 2011), while lowlands receive on average of 440 mm (Taita Taveta County Government, 2015). As the rains come from the east, the eastern part of the hills receive more rain, while the western parts remain in a rain shadow. Consequently, the highland area has a denser vegetation and more favourable conditions for agriculture. The lowlands' dry climate (Figure 9) has resulted in more livestock farming and less crop production.

Figure 9. Water Scarcity in Taita region.

The lowland has arid and tropical savanna climate while the highland with more abundant rainfall has a monsoon climate due to the orographic effects. The average temperature in the highland is about 17 °C compared to about 25 °C in the lowland. There has been a depletion in the water availability in the area, as a result of population growth and climate change (Kivivuori, 2013; Hohenthal et al., 2014). The vegetation in the area has also been depleted over the years, leaving little forest fragments in the mountain tops. Dry forests and woodlands can be found in the midlands at about 1200 meter above sea level (Krhoda, 1998: 32). The lowland area is largely

covered by bushland and thicket, with a few trees distributed sparsely (Jaetzold & Schmidt 1983: 248). Deforestation has further increased the vulnerability of the area to erosion (Hermunen et al., 2004).

The isolation of Taita Hills has made it as a habitat to various endemic plant and animal species and it is considered as one of the world's 25 biodiversity hotspots (Myers et al., 2000). The common crops found in Taita Hills include peas, maize, banana, beans, tomatoes and so on. The farming techniques are mainly manual, with the use of hand-held hoes, cutlasses and animal-driven ploughs. These crops are mostly terraced, as are other parts of East-African highlands (Soini, 2006). There are also reserved fields for animals to graze, with the locals rearing mainly chicken and cattle. Figure 10 shows some other water uses in Taita.

Figure 10. Water is used for irrigating fields, for animals and other livelihoods like making bricks.

According to the 2009 census, the population of the county was 284,657 and there has been an annual growth rate of 1.6 % between 1999 and 2009 (KNBS, 2010). Agriculture is the predominant rural occupation and largely small-scale and rain-fed. It mainly involves the cultivation of small-scale vegetables and crops for the local markets (Hohenthal et al., 2015). About 57.2 % of the people live below poverty line of KSH 1562 (15 USD) per month (Taita Taveta County Government, 2013). Furthermore, rooftop rainwater harvesting is used in many households to supplement other sources.

Figure 11. Unclean water being used for domestic purposes in Taita Taveta County.

The competition between wildlife, livestock and the people has led to the decline of the water resources. About 75 % of the people do not have access to potable water supply, as shown in Figure 11. (Taita Taveta County Government 2014). This can also be attributed to the deforestation, as the "water towers" are on the decline. These "water towers" are cloud forests that are found in the forested highland and help in generating moisture from fog (Hohenthal et al., 2015). The highland area around Wundanyi catchment area are densely populated, mostly cultivated and least water scarce while the lowlands are suffering from water shortages.

4 DATA

4.1 LiDAR data

The Optech ALTM 3100 sensor was used to obtain the LiDAR data in the flight campaign in February 2013. The sensor recorded the first, last and two intermediate pulses. More information can be found in Table 2. The second campaing was carried out using Leica ALS60 over a period of one year with several flight campaigns (Table 3).

Table 2. Parameters of sensor and settings of the 2013 LiDAR data.

Parameters	Configuration settings
Date of Acquisition	February 2013
Sensor	Optech ALTM 2100
Maximum can angle	± 16°
Pulse frequency (KHz)	100
Scan frequency (Hz)	36
Average point density (pulses m^{-2})	9.7
Average return density (return m^{-2})	11.4
Average flying height (m)	750
Beam divergence at $1/e^2$ (mrad)	0.3
Average footprint diameter (cm)	22.5
Coordinate system	UTM zone 37S (WGS 1984)

Table 3. Parameters of sensor and settings of the 2015 LiDAR data.

Parameters	Configuration settings
Acquisition Dates	26th January, 6th & 8th February, 2014 & 5,6,11 & 13 February, 2015
Sensor	Leica ALS60
Maximum scan angle	± 16°
Pulse frequency (KHz)	58
Scan frequency (Hz)	66
Average range (m)	1460
Average point density (pulses m^{-2})	3.1
Average return density (returns m^{-2})	3.4
Range of Pulse density (pulses m^{-2})	1.0-4.9
Beam divergence at $1/e^2$ (mrad)	0.22
Average footprint diameter (cm)	32
Coordinate system	UTM zone 37S (WGS 1984)

4.2 Rainfall data

The mean monthly and mean annual rainfall datasets were acquired from Climatologies at high resolution for the Earth land surface areas (CHELSA). The data was obtained from CHELSA's project's website. CHELSA is a high resolution (30 arc sec, ~1 km) climate data set for the land surface areas of the Earth that incorporates topoclimate (e.g. orographic rainfall & wind fields). It consists monthly and annual average patterns of temperature and precipitation from 1979 till 2013. CHELSA_v1 was derived by statistically downscaling the ERA interim global circulation model, with a GPCC and GHCN bias correction. (Karger et al., 2017). ERA-Interim is a global atmospheric reanalysis from 1979 which is updated in real time continuously. GPCC analyses global precipitation and is utilized for monitoring and studying the global climate. The centre was contributed by Germany to WCRP and GCOS. GHCN-M is a temperature dataset which was first developed in the early 1990s, with second and latest versions released in 1997 and 2011 respectively.

The CHELSA rainfall data are in a geographic coordinate system referenced to the WGS 84 horizontal datum, in which the horizontal coordinates are in decimal degrees. The layer extents of CHELSA are due to the coordinate system which derived from the 1-arc-second GMTED2010 data. This also inherited the extent of its grid from the 1-arc-second SRTM dataset. Due to the pixel center referencing the input data - GMTED2010, the full extent of every CHELSA grid (outside the vertices of the cells) is different from an integer value of longitude or latitude by 0.000138888888 degree (or 1/2 arc-second).

4.3 Survey data

A qualitative approach was used by having households survey to get insight into the water situation in the area. The fieldwork took place from 2nd August until 13th August 2018. Due to the rough terrain of the area, the fieldwork had to be done by sampling houses along the roads in various parts. I also tried to reach some houses away from the roadside. To ensure quality response, I worked with a field assistant who is a local and was able to communicate in the local kitaita language. 120 questionnaires were administered, out of which 105 were recovered. 46 households were sampled in the lowlands while 59 were sampled in the highlands.

4.4 Other dataset

Rainfall datasets were obtained from ground weather stations managed by Taita Research Station in 6 different locations in the study area. The data existed separately and had to be aggregated and cleaned by Python 3.7. Google satellite imagery was used for digitizing the buildings which were used to validate the rainfall model obtained from CHELSA.

5 METHODOLOGY

5.1 Software and Workflow

The processing of the LiDAR data was done in with *lastools* in the windows command line. All the analysis of the RRWHP modelling and data cleaning were done in Python 3.7. The development environments used were the *spyder* IDE and visual studio code editor. Some of the libraries used include, *numpy, scipy, pandas, geopandas, shapely, rasterio, gdal, fiona, matplotlib* etc. Aside the study area map, all other maps were created and visualized using Python also. The survey data was analyzed and visualized in SPSS and Microsoft Excel. *Git* and *github* were used for the version control. Figure 12 shows the entire work flow in creating my model

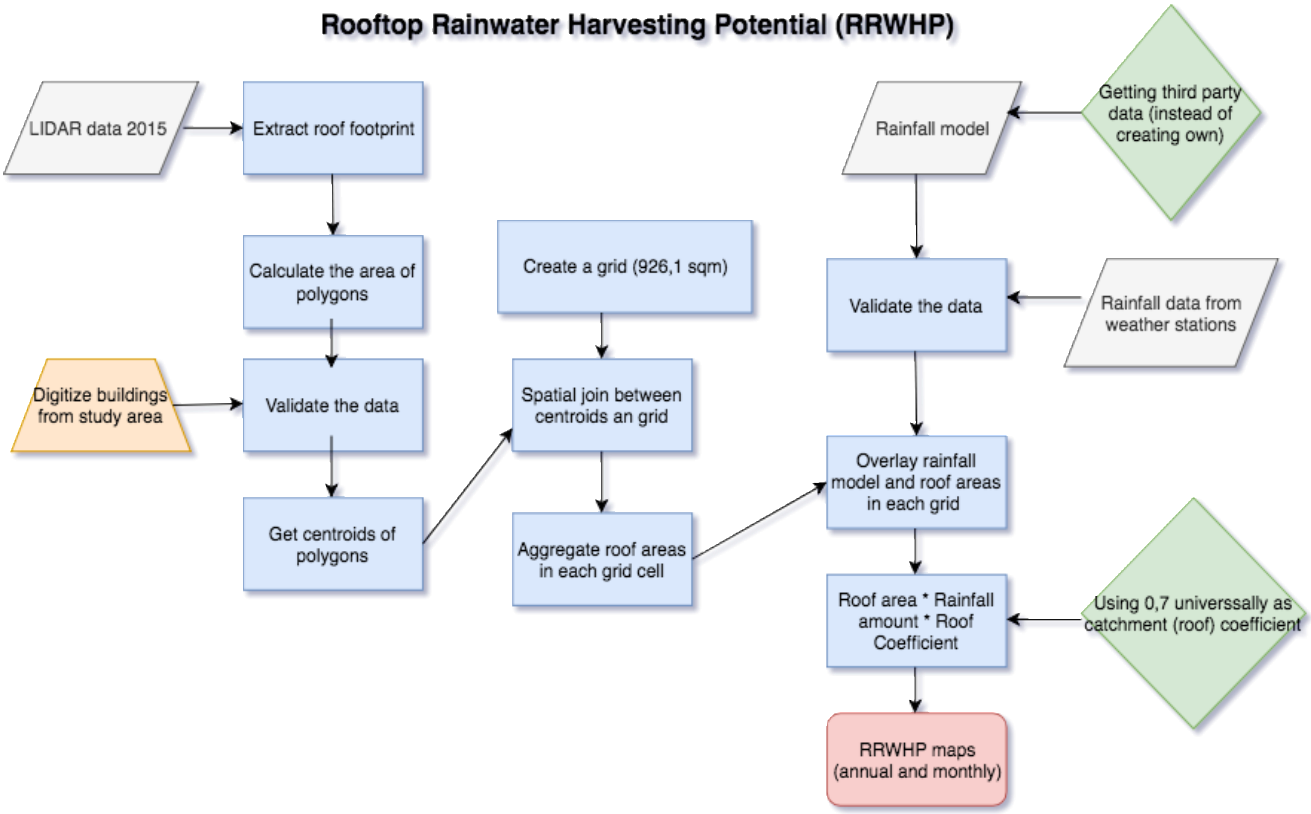


Figure 12. Conceptual framework of the whole process.

5.2 Roof/Building Extraction

The average point density was 11.41 points/m^2. The average footprint was 22.5 cm. The second LiDAR data was obtained in June 2015, also, with first, last and intermediate pulses. However, the average point density was 3.27 points/m^2, which was much lower than that of 2013. The raw output point cloud was already georeferenced. The quality of the data was checked with *lastools*, by using the *lasinfo* command. I received the data as multiple tiles which I aggregated in Python 3.7 to check the quality and see the distributions (Figure 13 and Figure 14). Other quality checks were done with other commands such as *lasoverage* to see overlapping flight lines. Table 4 and Table 5 show the point density and spacing of all returns and last returns of 2013 and 2015 dataset, respectively. The point density of all returns point cloud range from about 6 points/sqm to about 12 points/sqm average of 11 points/sqm and most grids having about 10 points/sqm.

Table 4. Pulse density and point spacing of 2013 LiDAR data.

		mean	***STD***	***min***	***25%***	***50%***	***75%***	***max***
All returns	**Density**	11.41	3.87	6.08	9.14	10.27	12.23	36.01
	Spacing	0.30	0.04	0.17	0.29	0.31	0.33	0.41
Last returns	**Density**	9.66	2.96	5.42	7.83	8.80	10.61	27.28
	Spacing	0.33	0.04	0.19	0.31	0.34	0.36	0.43

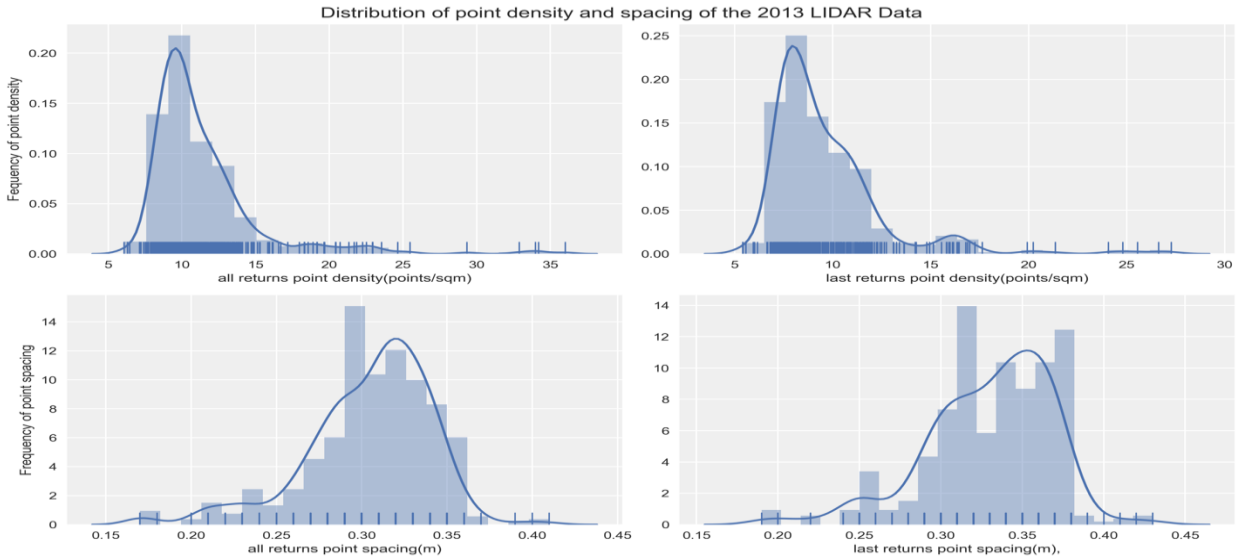


Figure 13. Point density of 2013 LiDAR data.

On the other hand, the point density of the LiDAR data of the 2015 campaign ranges from about 1 to 5 points/sqm with average and most tiles having about 3points/sqm. The point spacing, however ranges from about 0.5 m to 0.5 m and average of about 0.6 m (Table 5).

Table 5. Pulse density and point spacing of 2015 LiDAR.

		mean	***STD***	***min***	***25%***	***50%***	***75%***	***max***
All returns	**Density**	3.27	0.86	1.41	2.58	3.41	3.86	4.93
	Spacing	0.57	0.08	0.45	0.51	0.54	0.62	0.84
last returns	**Density**	2.93	0.72	1.30	2.32	2.96	3.56	4.47
	Spacing	0.60	0.08	0.47	0.53	0.58	0.66	0.88

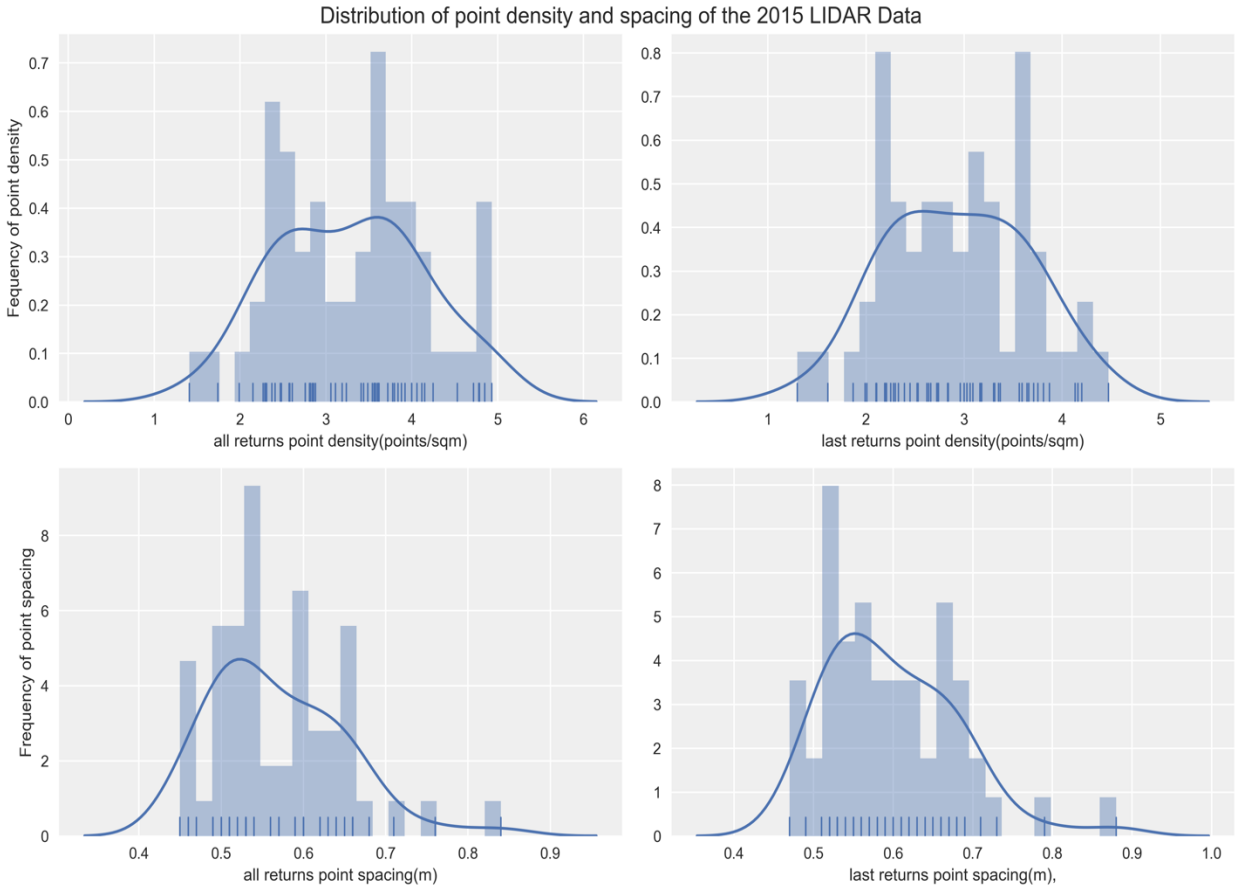


Figure 14. Point density of 2015 LiDAR data.

Figure 15 illustrates the workflow for the automatic extraction of buildings' footprints from point cloud of LiDAR data generated in 2013 and 2015. The qualities of the data were compared afterwards.

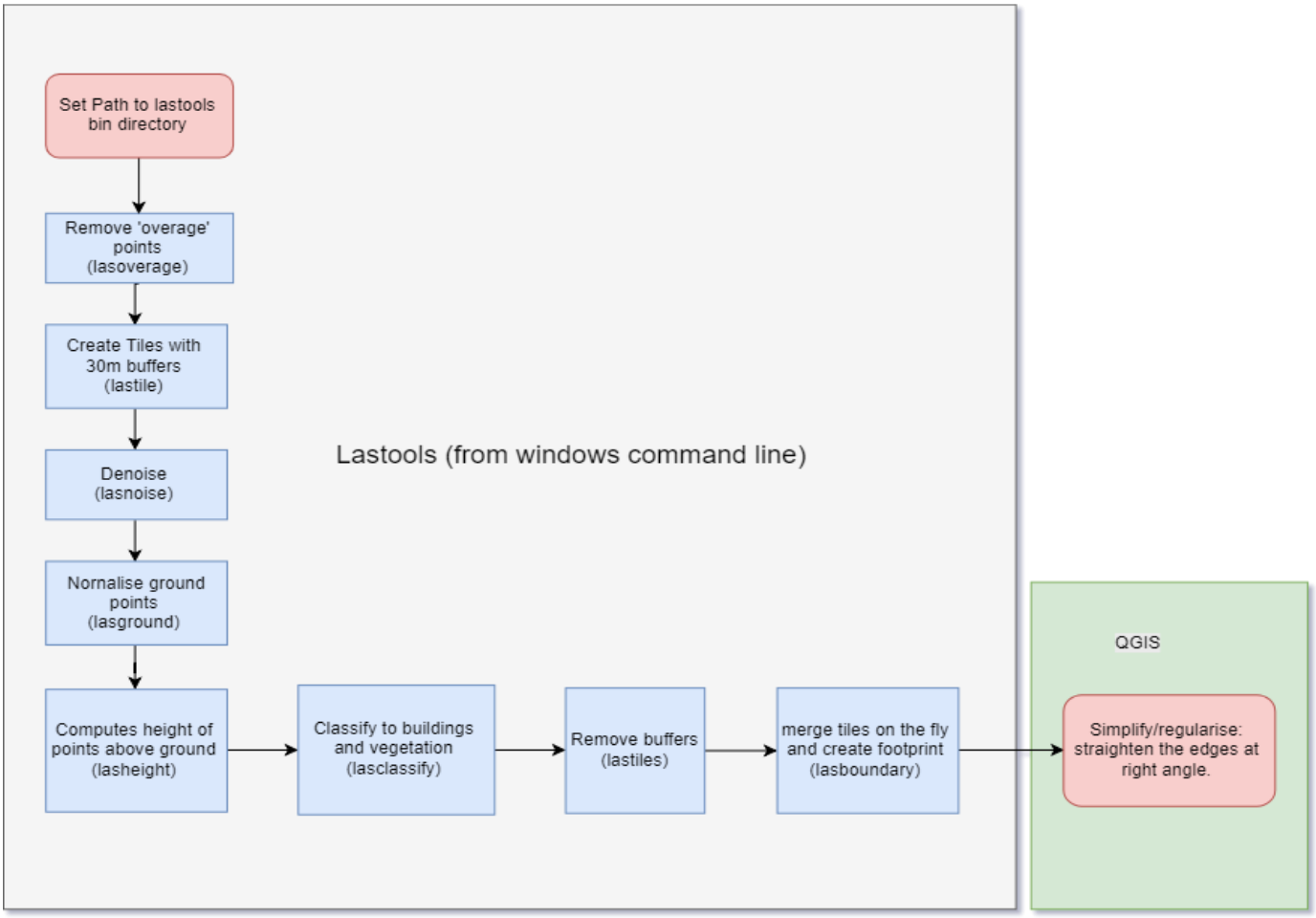


Figure 15. Work flow of generating the buildings from LiDAR data.

I started by checking the quality of the data (Figure 16). The point density and point spacing are two important factors that affect the quality of LiDAR data.

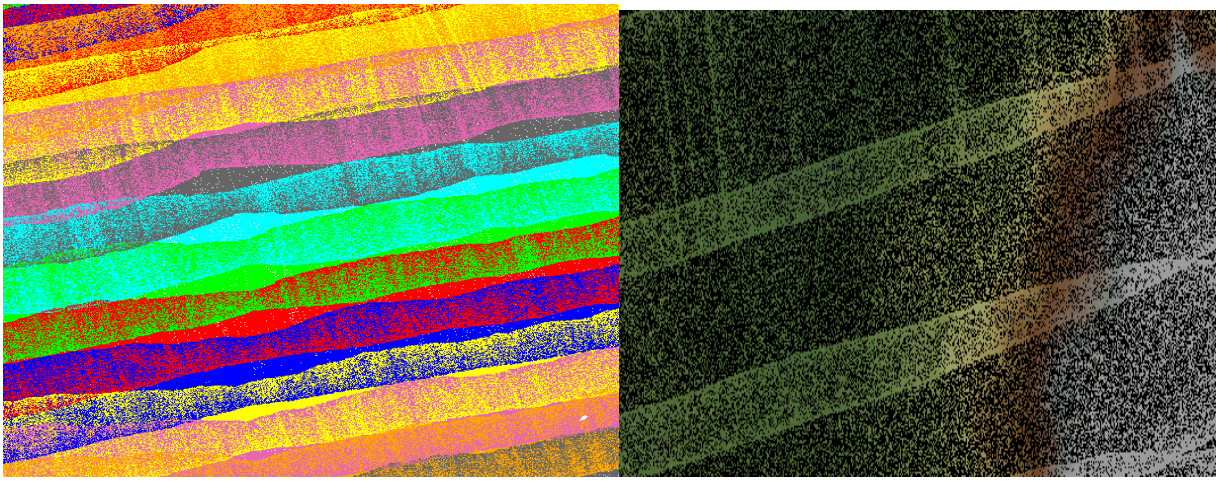


Figure 16. Overlapping LiDAR flight lines in Taita Hills.

After all quality checks, I removed the coverage points that got covered by more than one flight line. This was done with the *lasoverage* command. With the LiDAR data in flight lines, I used the *lastile* command to create square tiles of size 300 by 300 meters. This was done to lower the number of points per file and for memory efficiency and to speed up the computation. A buffer of 30 meters was added to every tile to prevent edge artifacts in further processes. 30 meters was chosen as it was expected to be larger than all the buildings in Taita Hills and bigger than the step size used later in the classification of the points. Thereafter, I classified the noise points into class code 7 which is the default code. This was done to remove isolated points which have five or less points in their surrounding of 4 by 4 by 2 meters grid of cells. Thereafter, the data was visualized based on the various returns (Figure 17).

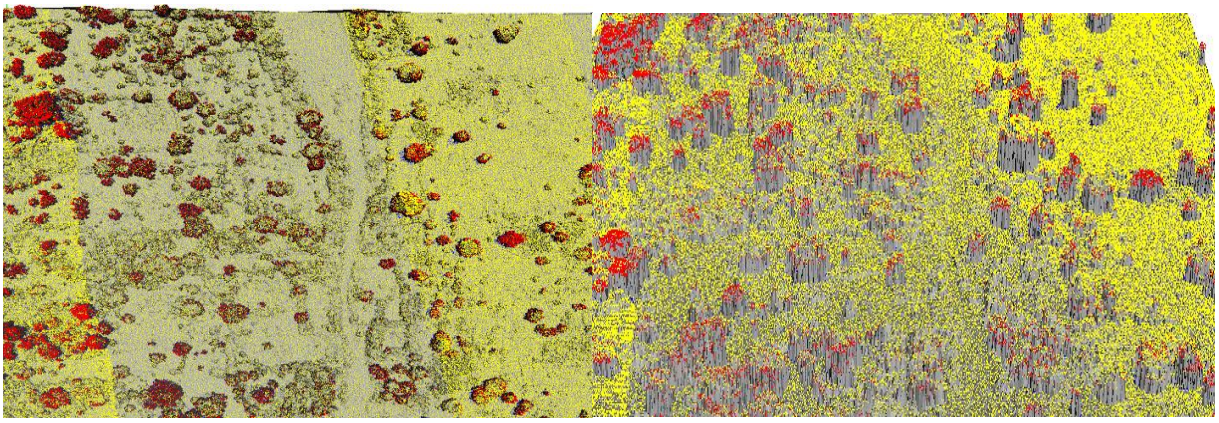


Figure 17. A triangulated tile showing points based on returns.

Next step was the extraction of bare-earth by classifying the denoised tiles into ground-class 2 and non-ground points - class 1. I ignored the class 7 - noise points. The coarseness was set as fine (Figure 18).

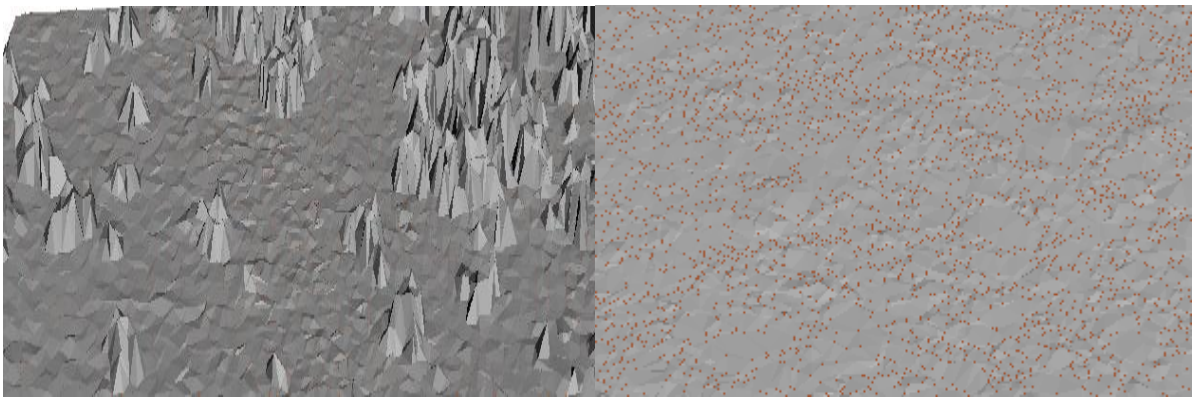


Figure 18. Triangulation of the ground point of 2015 LiDAR data.

Next, I computed the height of each of the LiDAR point above the ground which was generated earlier. Height above 40m and below 2 meters were dropped because of the expected height of buildings in the Taita Hills. Afterwards, I used *lasclassify* command to further classify the non-ground points - class 2- from the above into building- class 6, and high vegetation - class 5. The classification of each tile was checked, and some misclassifications were manually corrected. The green patches show the vegetation and the two little patches of orange are the buildings. The black points are the unclassified points (Figure 19).

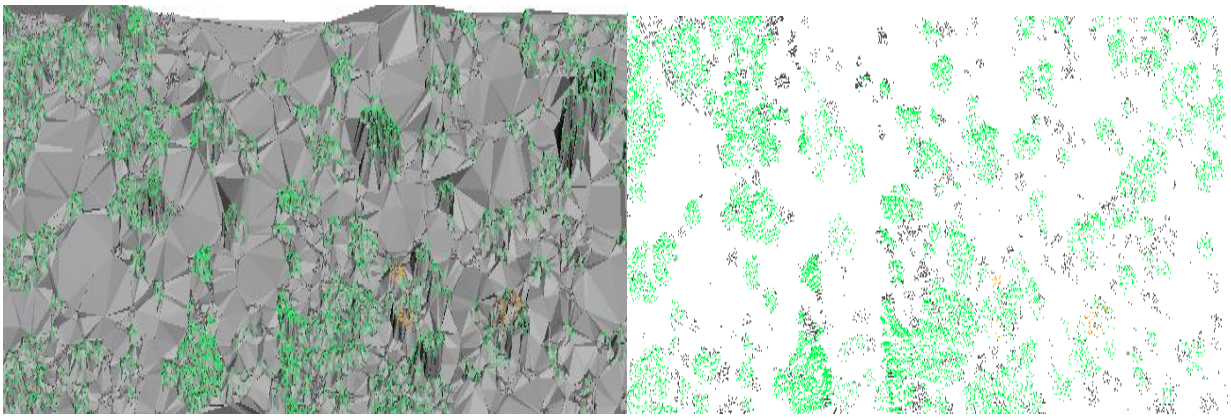


Figure 19. Classification of vegetation and buildings in Taita Hills.

The buffers created earlier which have been used for computations thus far, were then removed to prevent artifacts along tile boundaries. This was done with the *lastile* command. DEM and DSM are popular derivatives of LiDAR data (Figure 20). This was also generated for visualization purpose by interpolating the ground points with a triangulation. Thereafter, the tool sampled the TIN at the center of every cell. I computed the DEM and DSM from the ground and first returns respectively, using the *las2dem* command. The *"use_tile_bb"* parameter was used to limit the generation of the raster to the 300m^2 tiles, so that it does not rasterize the 30 m buffer too. The file was then imported into ArcGIS to create a hill shade with a striking visualization.

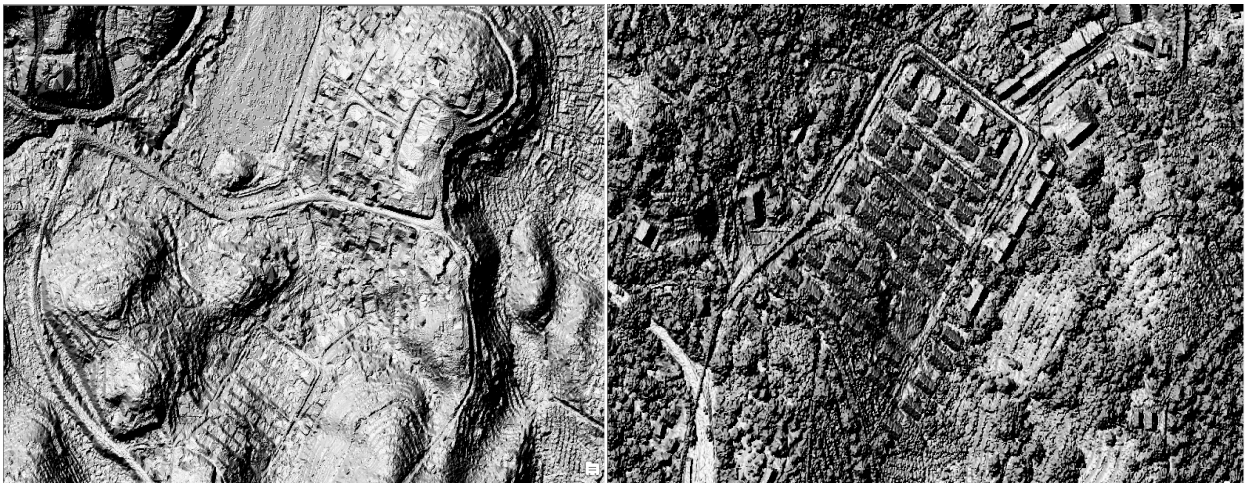


Figure 20. On the left is a DEM, and on the right is a DSM over Wundanyi town, Taita Hills.

Lastly, I generated the building footprints by merging the classified tiles on the fly and keeping class 6. *Lastools* is scalable and capable of handling this and prevents duplicate points from the buffer with less accurate classifications. The *lasboundry* command was used for this. Disjoint and concavity of 1.5 were chosen to prevent close buildings from joining together. After generating the building footprints, there were simplified in QGIS with tolerance of 1.5. The area based "Visvalingam" algorithm was used for this.

5.3 Polygon Simplification: Visvalingam–Whyatt algorithm

The buildings were assumed to be rectangular and orthogonal. Although, this hypothesis may be too simplistic or restrictive for urban areas, similar approach is utilized in this research for the sake of simplicity and also bearing in mind that the study area is rural. A rectangle is reasonable

geometric shape descriptor which can also be combined to derive complex structures. Studies have used Douglas-Peucker's technique (Douglas, 1973) for approximation of the contour of polygons' edge. In this study, I used the Visvalingam's technique in QGIS for the regularization of buildings' edges. It has been shown that Visvalingam–Whyatt algorithm has better performance for elimination of entire shapes, a term called caricatural generalization (Weibel, 1997) (Figure 21).

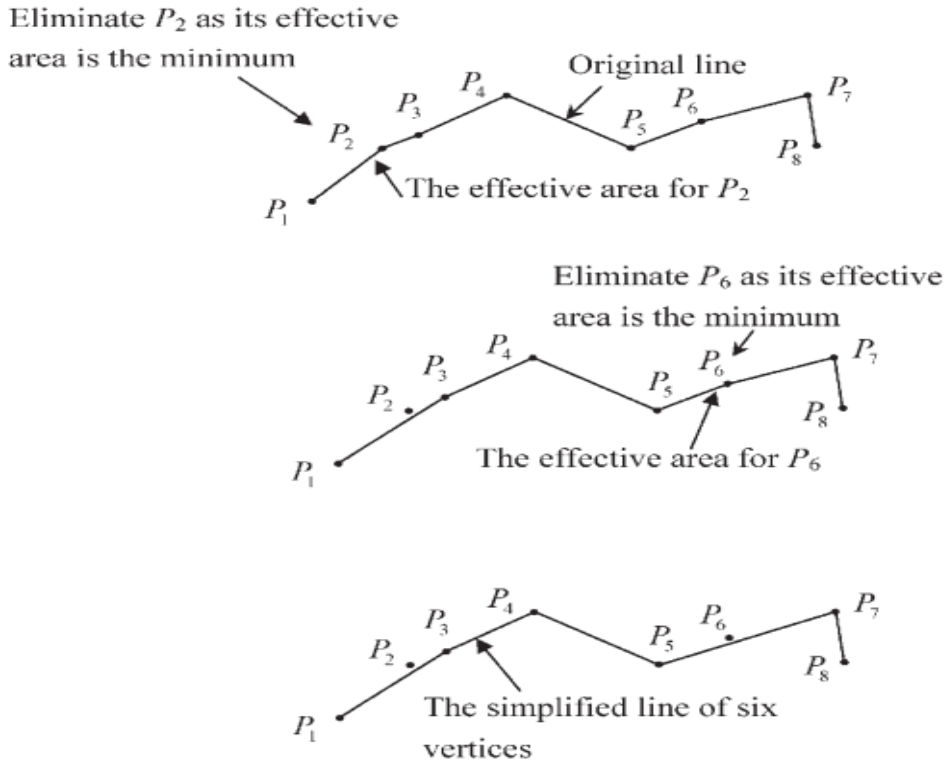


Figure 21: The Visvalingam and Whyatt (Shi & Cheung, 2006).

Visvalingam and Whyatt's (1993) algorithm is a global routine for the generalization of area-based line. The effective area for each original point between the start and end nodes of the original line is the triangle's area formed by itself and its immediate neighbors (Figure 21). Besides the starting and last endpoints, all original points' effective areas are first computed and eliminates the original point with the minimum effective area. The original point with minimum effective area and effective areas for two adjoining points to the point removed are recomputed at each iteration. It repeats this process until the predefined value (e.g. 2) is equal to number of points retained (Shi & Cheung, 2006). Figure 22 shows some simplified footprints of buildings in the area. After the extraction, I checked the outliers manually and with a boxplot. Based on this, I filtered out polygons smaller than $10m^2$ and those larger than $2{,}000m^2$.

Figure 22. Regularized roof polygons in in Wundanyi town, Taita Hills.

5.4 Runoff Coefficient

Runoff coefficient accounts for the surface type of the roof which determines the amount of rainfall that would be collected. The surface retains some of the water while others get evaporated. The rooftop coefficient for various types of rooftops shown in **Error! Reference source not found.**Table 6**.**

Table 6. Coefficient of various rooftop types (Dadhich, G., & Mathur, P., 2016).

S.no	Type of Rooftop	Rooftop coefficient
1	Galvanized sheets	0.90
2	Asbestos	0.80
3	Tiled	0.75
4	Concrete	0.70

A universal value was chosen for the roof coefficient because it was seen in the fieldwork that very few and negligible buildings have other kinds of roofs, aside galvanized iron sheets. Majority use this material because it is affordable and effective for rainwater harvesting. A value of 70 % was chosen because most roofs have a coefficient of about 70-80 % (Zhe Li & Fergal Boyle, 2010). As water harvesting is a crucial topic, it is important to reduce overestimation of the potential.

5.5 Rainfall Model

I had to decide between creating my own model by using the existing weather stations' data from the area or using CHELSA, which is a third-party already existing model. After cleaning and aggregating all the data from various stations, I decided not to use the weather station data because of low density and incompleteness of the data from two of the stations. Thereafter, the data was used for the validation of the CHELSA rainfall model. I utilized the monthly and annual precipitation data from CHELSA which is averaged over the period covers the period from 1979 to 2013. The first step was to clip the raster rainfall data to the area of interest. This was done using the *osgeo's gdal* tool in Python. Thereafter, the data were converted into vector format using *rasterio* library in Python.

5.6 Rooftop Rainwater Harvesting Potential

I calculated the areas of the footprints of the buildings and also derived the centroid of the buildings. I created a grid of cell size of 926.1 m^2 across the study area and clipped it to the study area. To get the data for every grid cell, an intersectional spatial join was carried out between the grid and the buildings' centroids. Centroid refers to the center of gravity of a polygon (Figure 23Figure 1). For a closed and non-self-intersecting polygon, the centroid is defined by n vertices (x_0, y_0), (x_1, y_1), ..., (x_{n-1}, y_{n-1}) is the point (C_x, C_y) (Bourke, Paul, 1997) where:

Equation 1.

$$C_x = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i\, y_{i+1} - x_{i+1}\, y_i) \text{ , and}$$

Equation 2.

$$C_y = \frac{1}{6A}\sum_{i=0}^{n-1}(y_i + y_{i+1})(x_i\ y_{i+1} - x_{i+1}\ y_i)\ ,$$

and where polygon's signed area (A), (Bourke, Paul, 1997) is:

Equation 3.

$$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i\ y_{i+1} - x_{i+1}\ y_i)$$

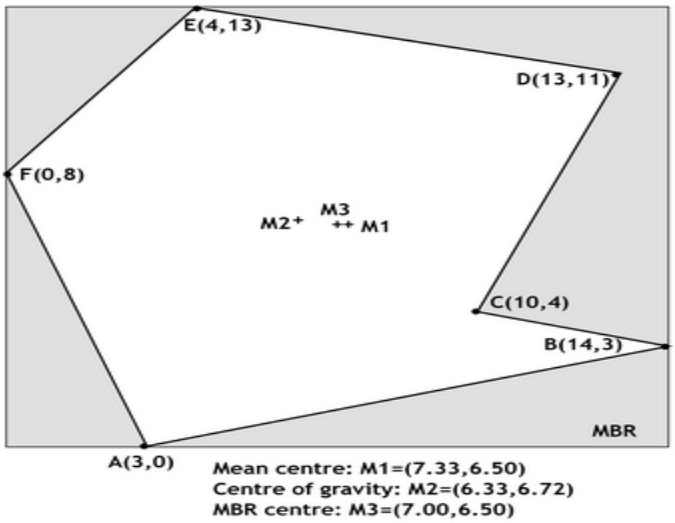


Figure 23. Centroid of a polygon (M2) and other kinds of centers of polygon's - mean center (M1), and center of Minimum Bounding Rectangle (M3) (De Smith, Goodchild & Longley, 2018).

The areas of all buildings in each grid were also summed. Furthermore, all the monthly and annual rainfall data were aggregated and joined to the grid spatially by intersection. Based on this, the total RRWHP was estimated using the below formula (Pecey, Amold and Cullis, Adrian , 1989):

Equation 4.

*RRWHP (litres) = Area of Catchments(m^2) * Runoff Coefficient * Rainfall (m)*

5.7 Rooftop Rainwater Harvesting Potential Satisfaction

In order to understand to what extent the RRWHP could satisfy the people's need, I estimated the water use per grid. This was based on the information derived from the field survey. This estimation was done by finding the product of number of buildings in each grid and the average daily water use in the area. This was then aggregated into monthly and annual value. Thereafter, the values were compared to the RRWHP of each grid by subtracting the values. This yielded the net satisfaction of the water use by the RRWHP. Furthermore, the values were classified as binary – positive and negative for the monthly and annual values.

Equation 5.

*RRWHPSat = RRWHP – (average water-use per household * number of households * time)*
Where:
RRWHPSat is Rooftop Rainwater Harvesting Potential Satisfaction,
** is the multiplication sign*

5.8 Accuracy Assessment and Validation

5.8.1 Validation for Buildings Footprint

Errors of omission and commission were derived to assess the accuracy of the extracted footprints of the buildings. An area was subset to compare the accuracy of the 2013 and 2015 LiDAR data with different point density and spacing. To achieve this, I manually digitized all the buildings in the area and ran an overlay analysis between the digitized and the automatically extracted buildings for 2013 and 2015. By comparing the total number of digitized buildings with the automatically extracted counterparts, I was able to get the number of omitted buildings in the process of extraction. The false negatives (i.e. omitted buildings) were derived as polygons that exist in the digitized buildings but not in the automatically extracted polygons. In other words, these were the digitized polygons that were not overlaid. The false positives were those that existed in the automatically extracted but not in the digitized.

True positive = overlaid

False Negative (error of omission) = total digitized polygons – overlaid digitized polygons
False Positive (error of commission) = total auto extracted polygons – overlaid auto polygons
producer's accuracy (%) = 100 – omission
Thus: producer's accuracy (%) = 100 – commission

5.8.2 Validation of Rainfall Model Data

Due to the short duration and incompleteness of the weather station data, I decided to aggregate all the data and then compare it with the rainfall model. This was done by carrying out an overlay analysis between all the station points and the rainfall grids. The monthly average rainfall was calculated for every station and then placed together in the same column and compared with the modelled monthly rainfall data from CHELSA. Based on this, the correlation and thereafter, the coefficient of determination was calculated between the modelled data and reference weather station data. The RMSE was also derived.

6 RESULTS

6.1 Building detection

For the entire study area, 94,632 polygons were generated after the extraction from LiDAR data. Subsequently, 38,323 polygons were left after filtering out buildings less than 10 m^2 and bigger than 2000 m^2. The number was reduced to 33,084 buildings after clipping to the study area. The areas of the roofs' polygons aggregated into the grid shows that most of the buildings are concentrated around the western part of the area. While some have less than 22 m^2 in total, some grids have well over 11,000 m^2 to 92,000 $m^{2.}$ (Figure 24). These areas would be suitable for rainwater harvesting.

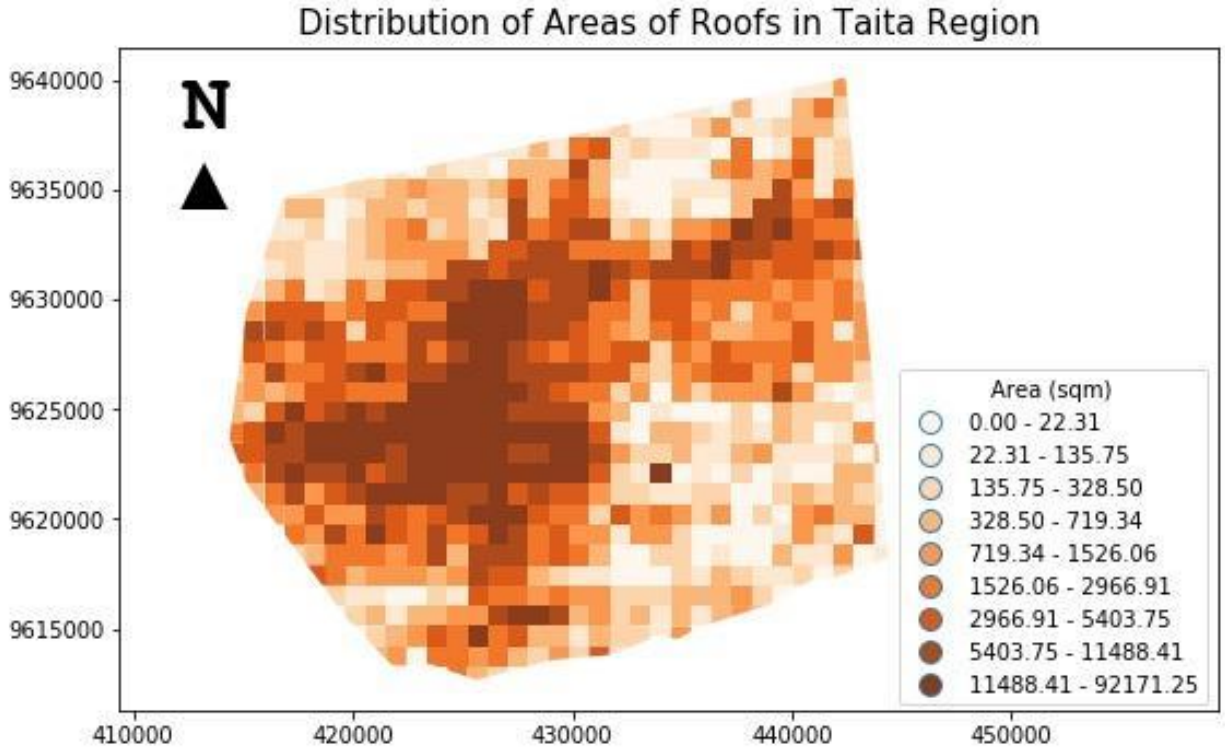


Figure 24. Spatial distribution of rooftop areas.

Determining the relationships of the overlaid digitized buildings and automatically extracted polygons can also give some insight into the accuracy of the extraction. In the best scenario, a one-to-one relationship is desired. This means that there is no erroneous merging of separate buildings' polygons. After clipping 2013 and 2015 data to a subset of the study area, 2,345

and 3,560 buildings were in the found for 2013 and 2015 data, respectively. The filtering criteria was then applied, which left 2,159 and 2,365 buildings for 2013 and 2015 data, respectively. In the validation of the building, there are many one-to-one relationships. These are polygons of digitized roofs, each of which matches exactly one polygon of the automatically extracted polygon. There were 1,112 polygons of such case in 2013 and 1,010 in 2015 (Figure 25Figure 24). In some situations, however, the automatically extracted polygons intercept with more than one polygon of the digitized.

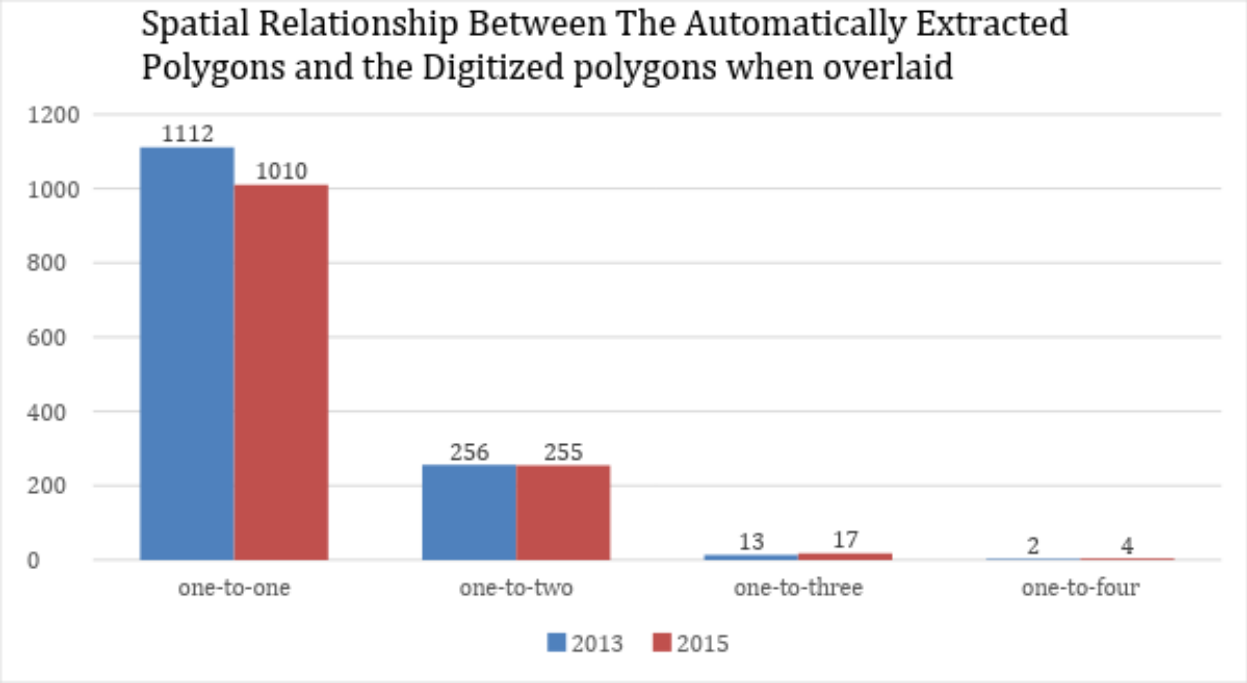


Figure 25. Validation of 2013 and 2015 datasets.

For 2013 versus 2015, there were about 256 versus 255 cases of one-to-two, 13 versus 17 of one-to-three and 2 versus 4 one-to-four relationships (Figure 25).

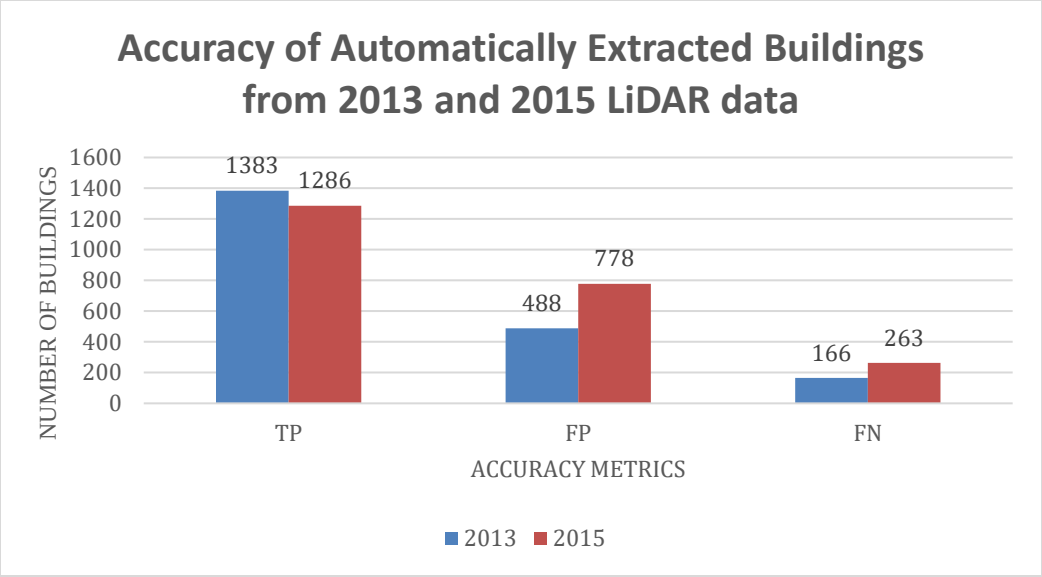


Figure 26. Accuracy of automatically extracted data from 2013 and 2015 data.

From the above, the 2013 data was more accurate in the detection of buildings. Figure 26 shows the monthly spatial distribution of rainfall in the area. There is more rainfall in November and December while June, July and August had the lowest rainfall amounts.

6.2 Rainfall Distribution in Taita

The rainfall model from CHELSA shows the pattern of distribution of rainfall in Taita with January and June to September being the dry months. Generally, the highland areas have more rainfall than the lowland. It shows that the mean monthly rainfall amount from June to July is less than 20 millimeters. April and November appear to be the wettest months, with some areas receiving up to 180 millimeters of rainfall in those months (Figure 27).

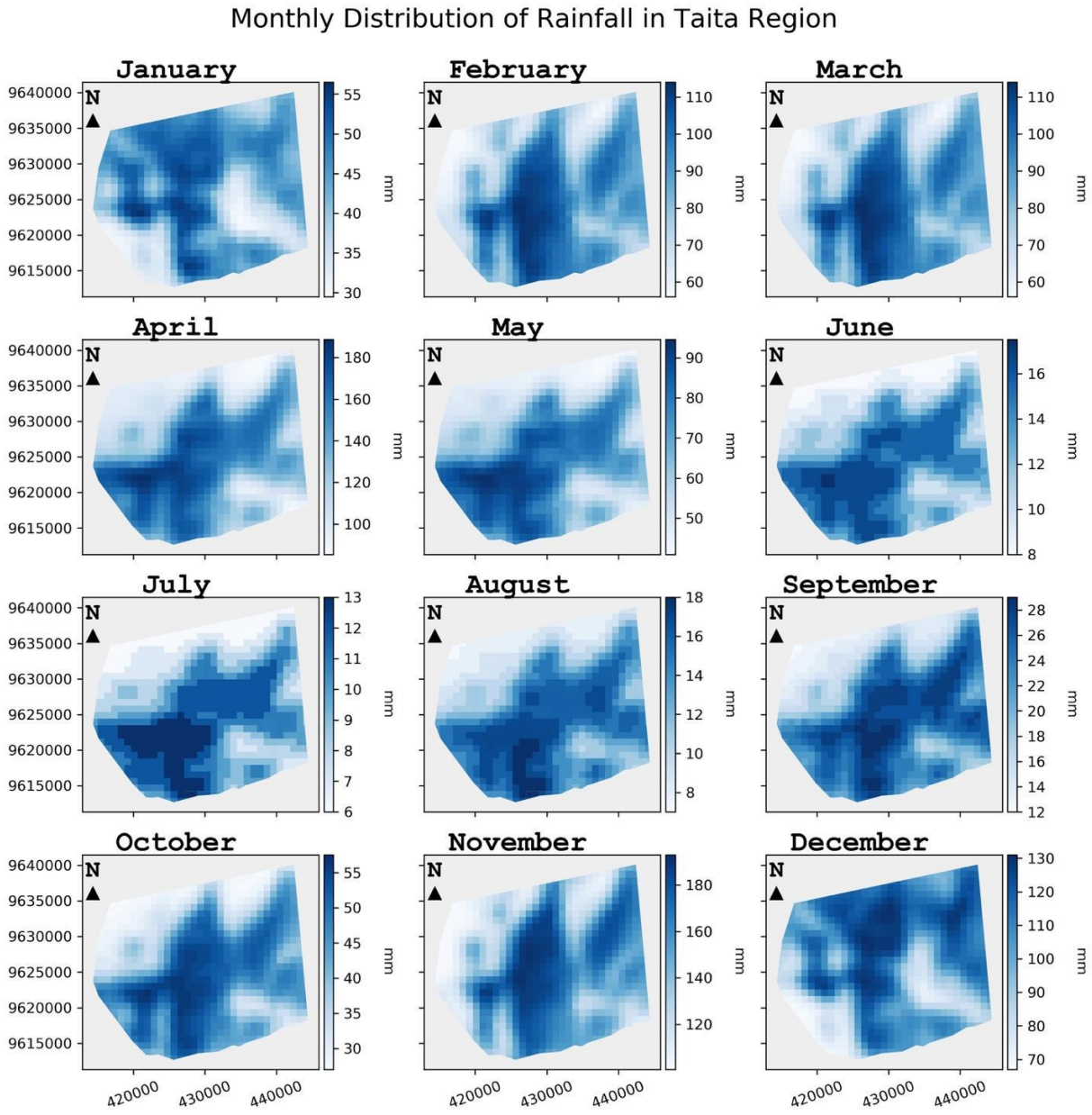


Figure 27. Monthly distribution of rainfall in Taita Hills.

Generally, the center to the western area receives the most rainfall in the highland while the lowland areas receive the least (Figure 28).

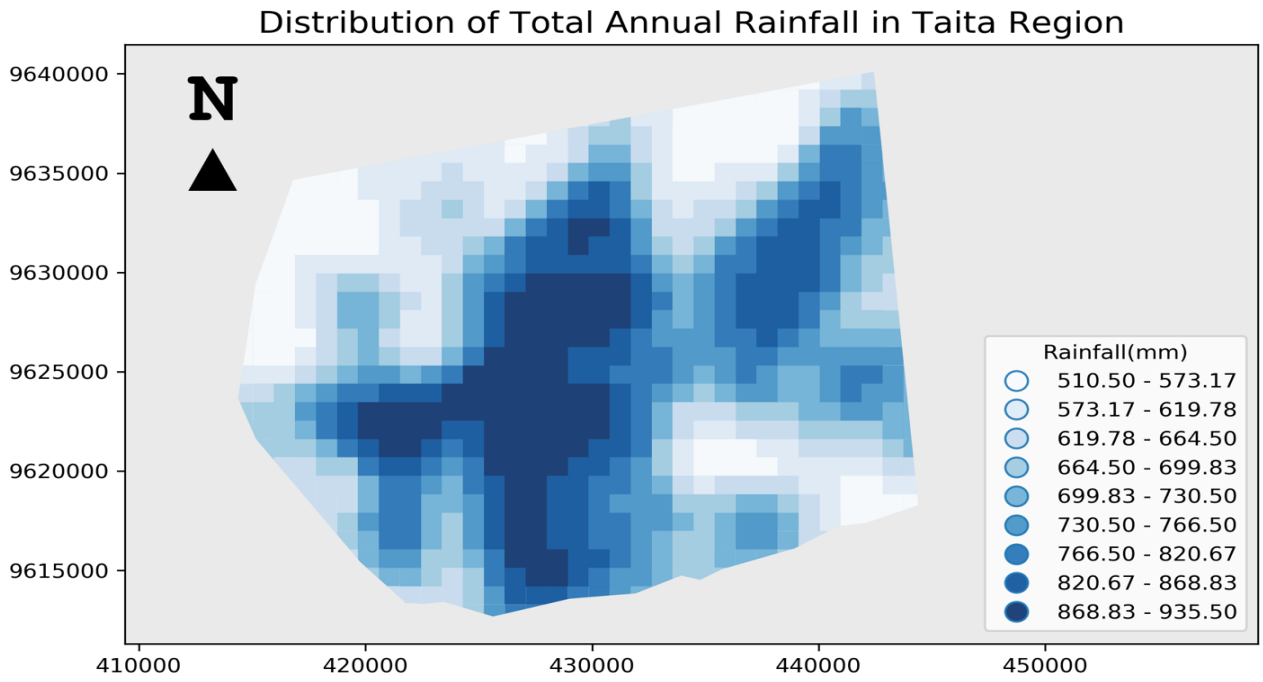


Figure 28. Mean annual rainfall distribution in Taita Hills.

Eight of the ground weather stations fall within the study area and mostly in areas with high rainfall (Figure 29).

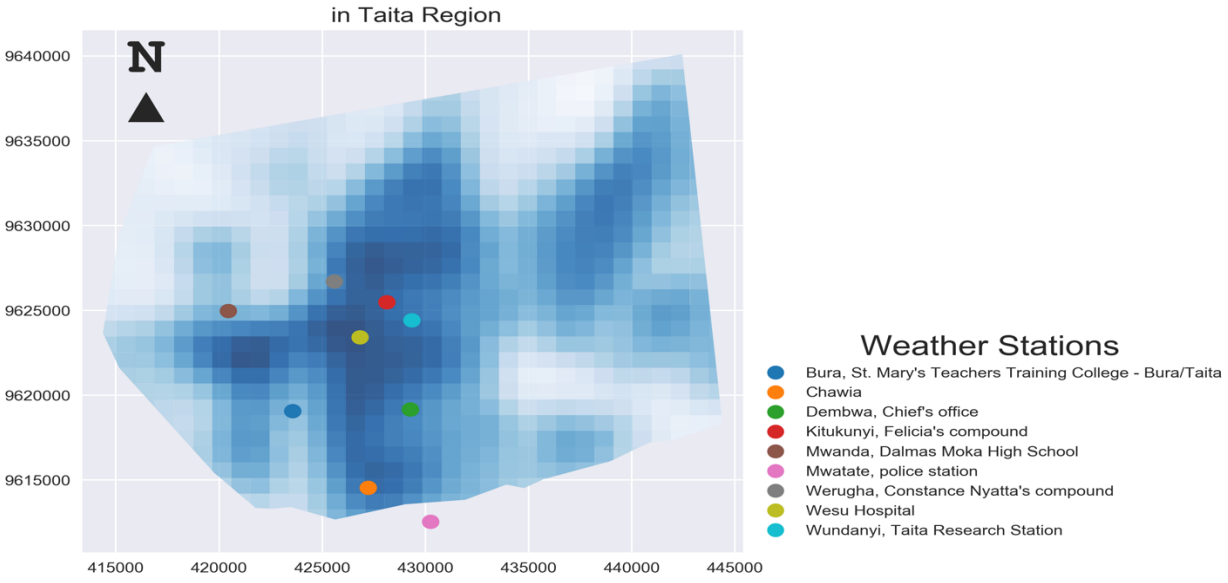


Figure 29. Locations of weather stations managed by Taita Research Station.

Stations with issues with the data were discarded, while the data from some stations were used. Figure 30 shows that Mwatate and Mwanda station had short-lived operations.

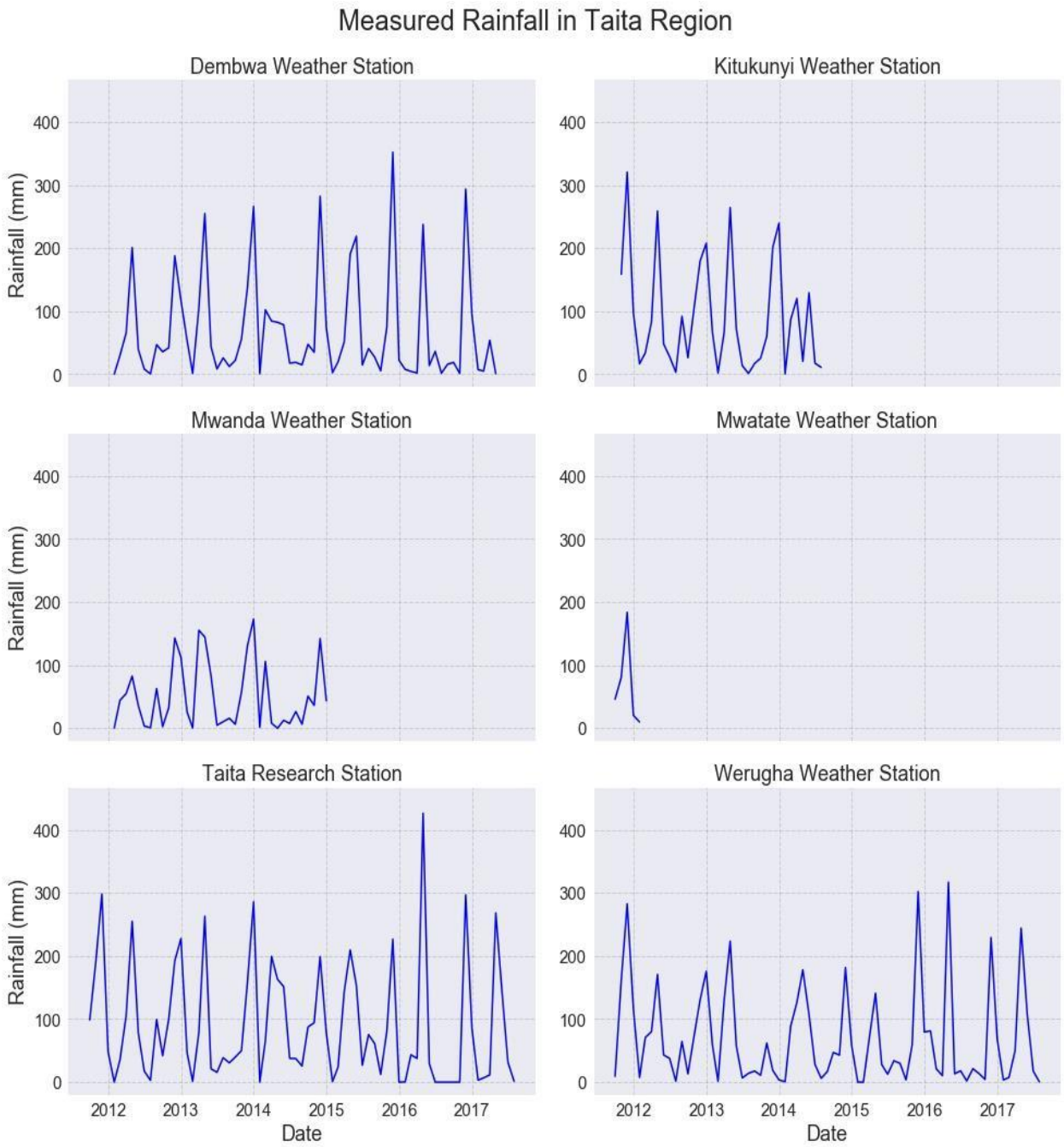


Figure 30. Measured rainfall data from the weather stations managed by Taita Research Station.

The rainfall model shows a good prediction when compared with the monthly average rainfall from the ground weather stations (Figure 31).

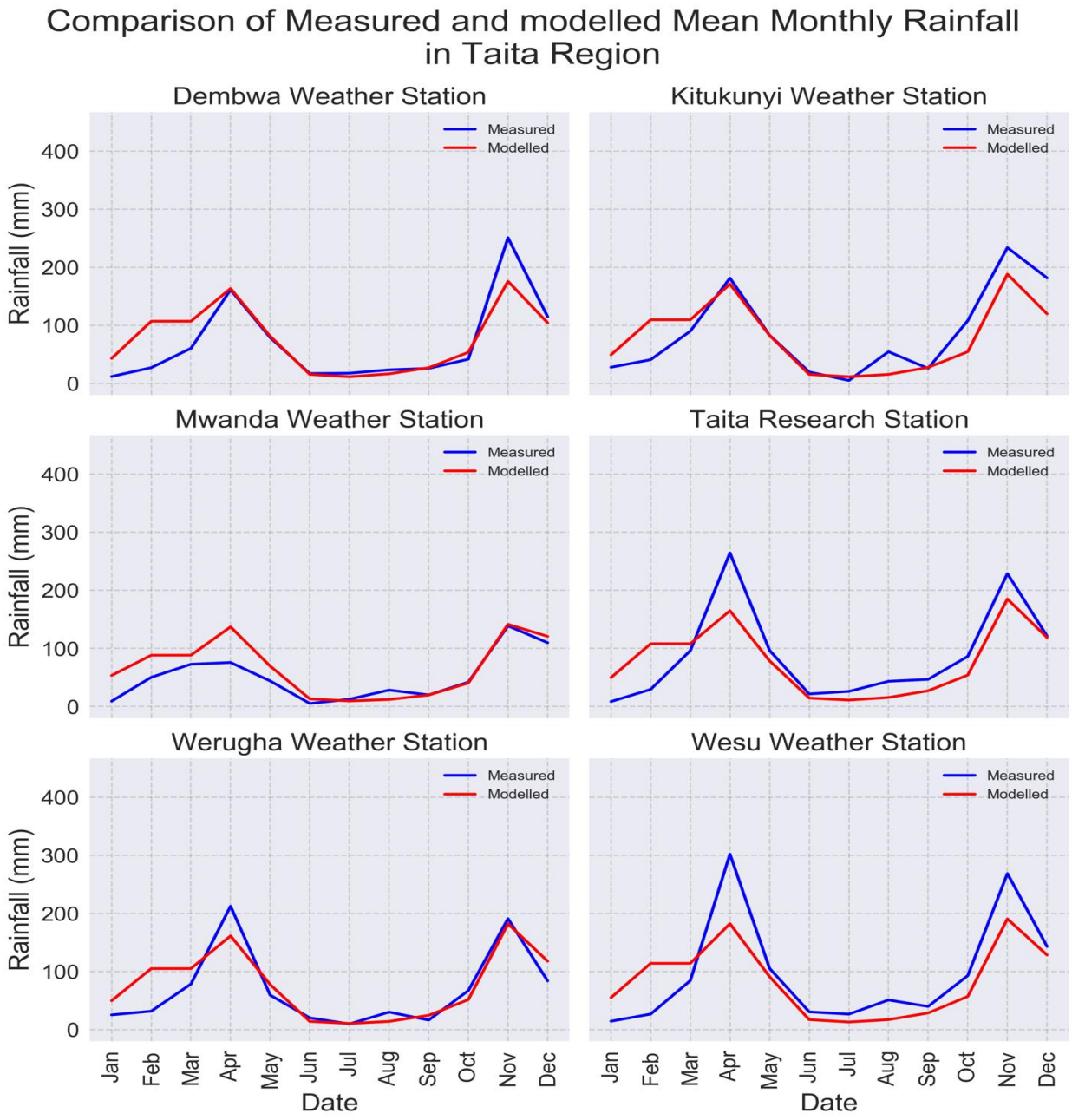


Figure 31. Comparison of measured and modelled rainfall data.

With further exploration, the modelled data shows quite good fit with the reference data from the ground weather station and with accuracy of about 72 % and RMSE of 38.74 millimeters (Figure 32).

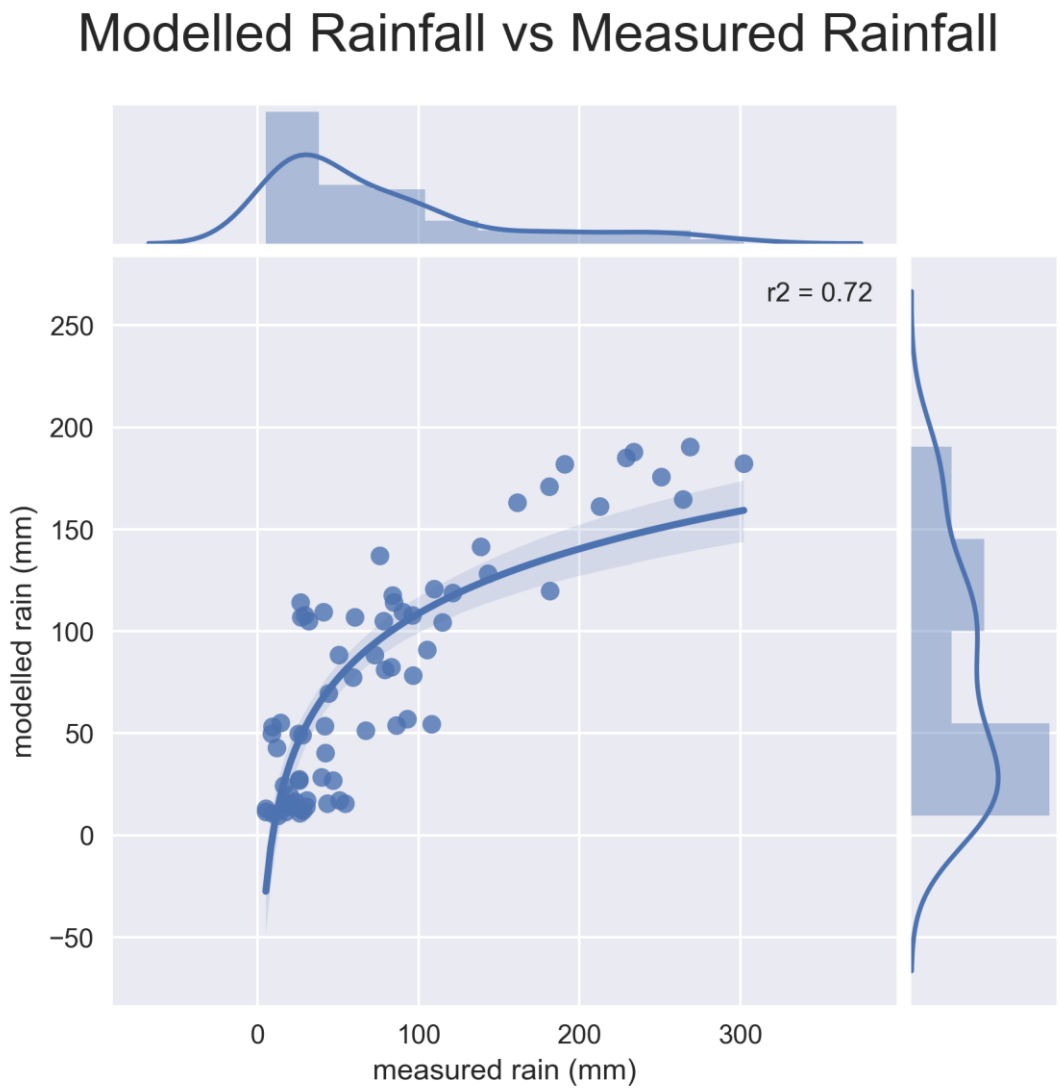


Figure 32. Validating modelled rainfall with measured rainfall.

6.3 Spatiotemporal Rooftop Rainwater Harvesting Potential in Taita

As the roof areas and rainfall are the critical factors that affect RRWHP, the high land area has more potential. The highland has more rainfall and more inhabitants. However, the RRWHP potential is very low across the area during dry period in June, July and August with only very few having rooftop water harvesting potential up to 500,000 liters (Figure 33).

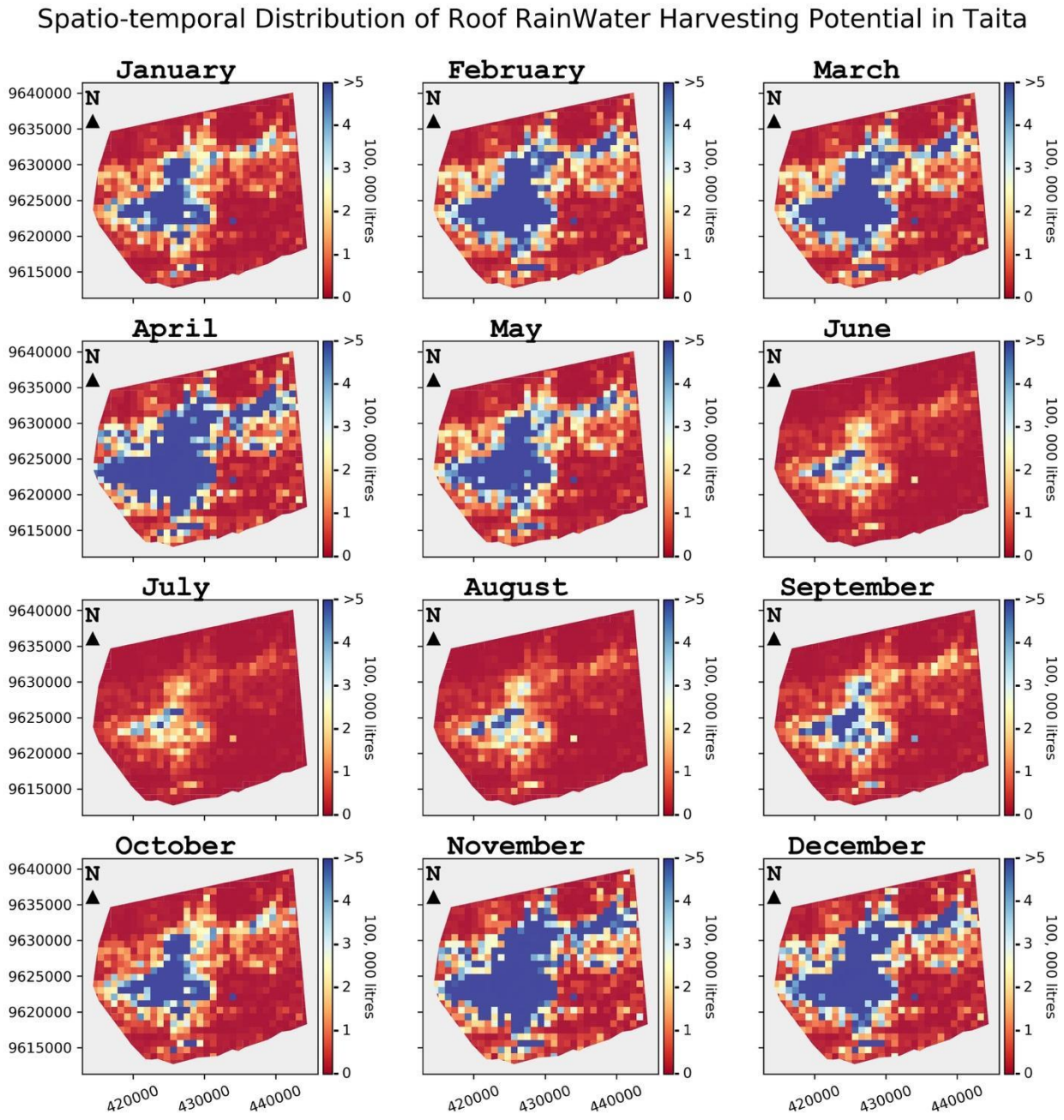


Figure 33. Spatiotemporal distribution of roof rainwater harvesting potential.

Figure 34 shows the high disparity of the annual RRWHP, with many areas of about 1sqkm having less than 9,000 liters while some have up to 6 million liters to 56 million liters RRWHP.

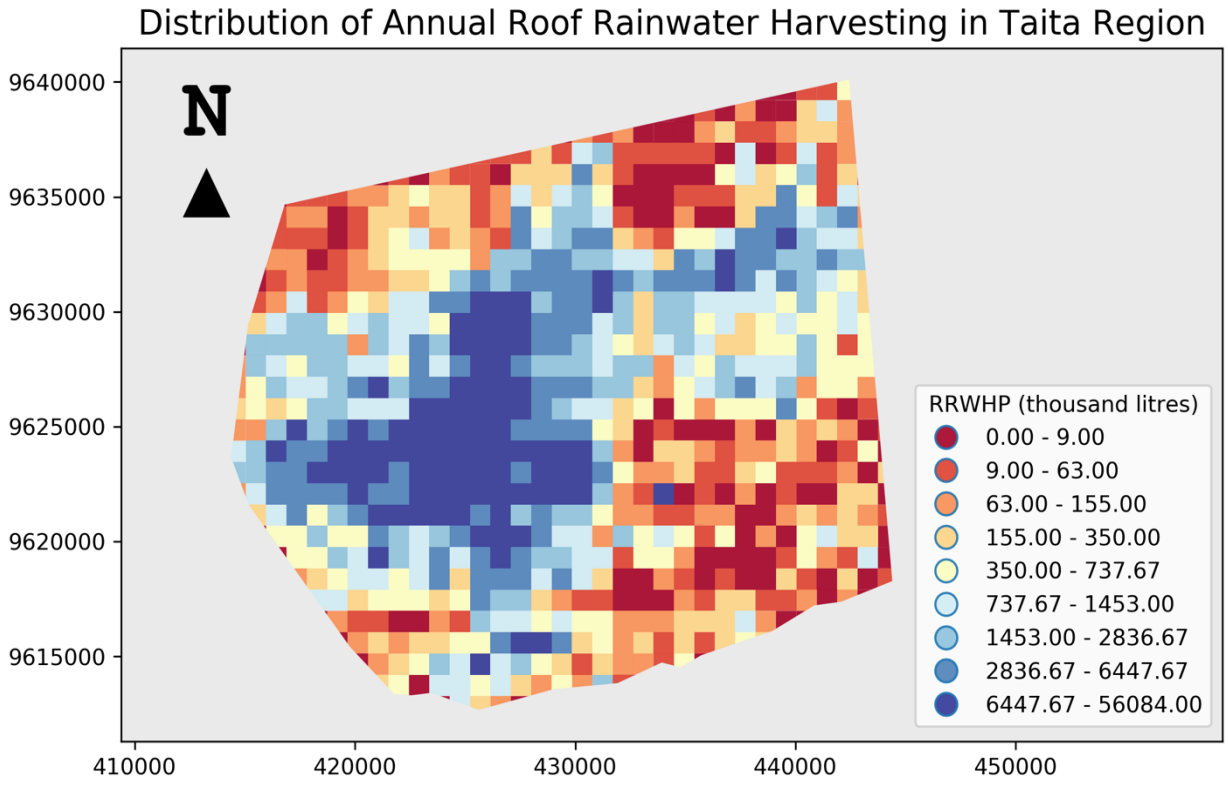


Figure 34 Distribution of annual roof rainwater harvesting potential.

Generally, there are two peaks which are in April and November, with about 400 million liters for the entire area. Conversely, the potentials are lowest in June, July, August to about 50 million liters or less (Figure 35).

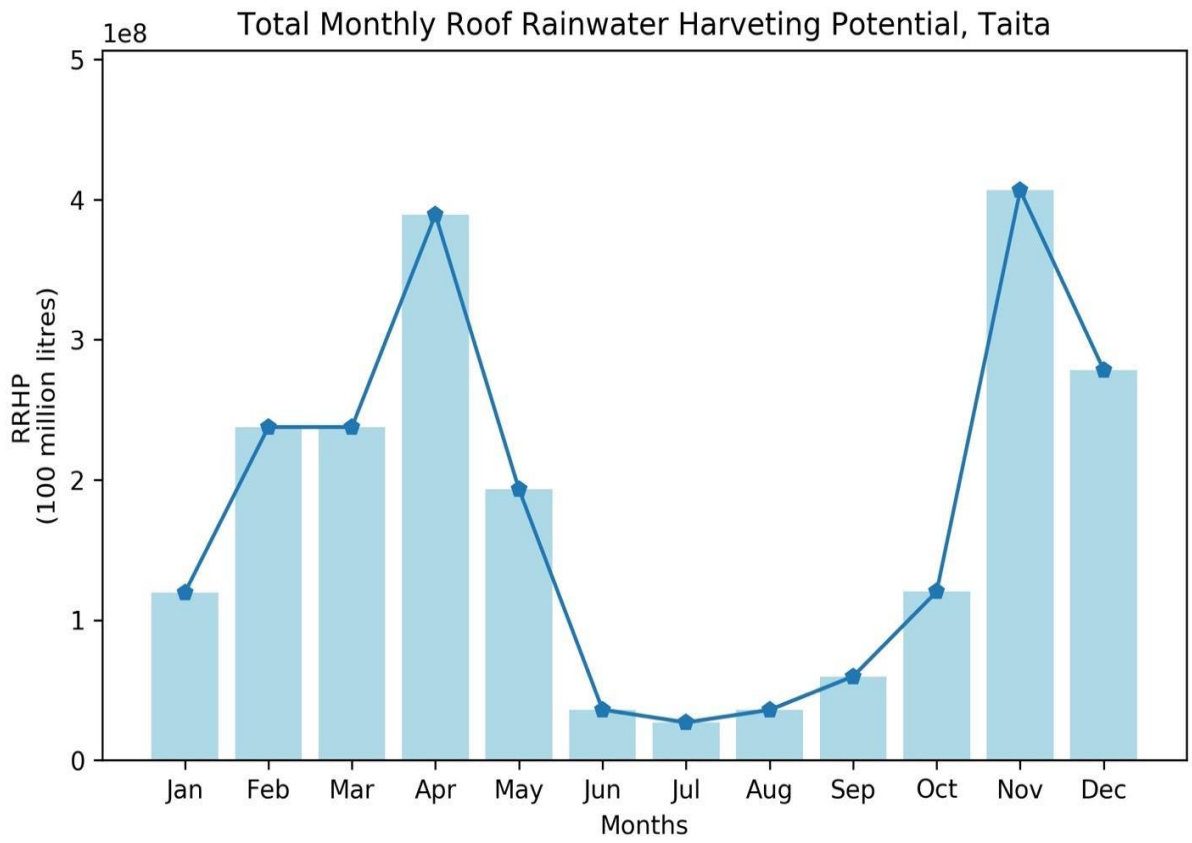


Figure 35. Monthly roof rainwater harvesting potential.

6.4 Potential water use satisfaction by RRWH

When the average water use per grid is estimated and compared with the RRWHP, it shows that the system alone cannot satisfy the water use of almost all the areas in the highland and lowland from June to September, except it is supplemented or stored from rainy months. Most of the areas have net positive values in April and November (Figure 36).

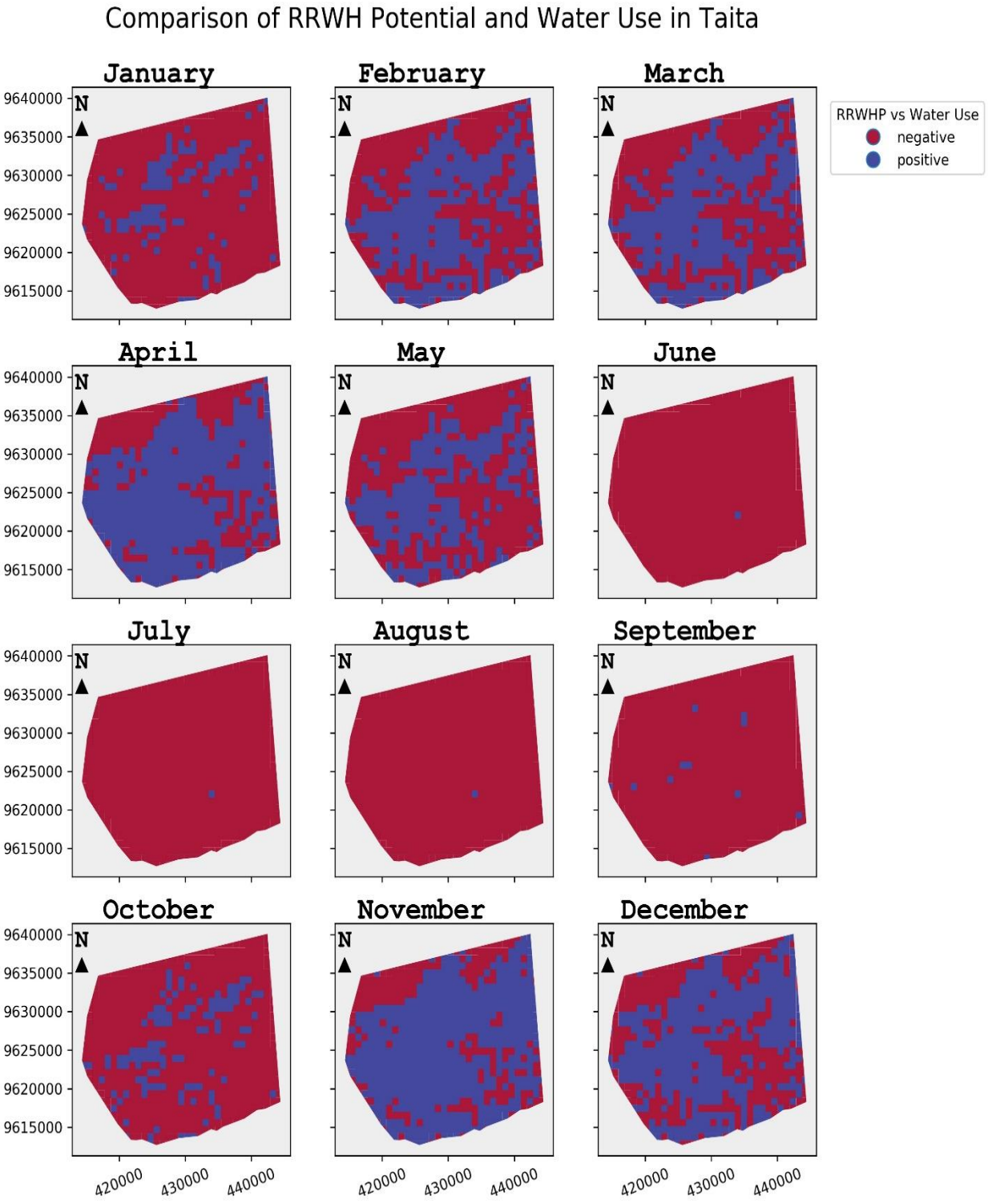


Figure 36. Potential satisfaction of water use by RRWH.

When investigated on the annual basis, it shows that some areas in the highland have huge deficit and (Figure 37 and Figure 38).

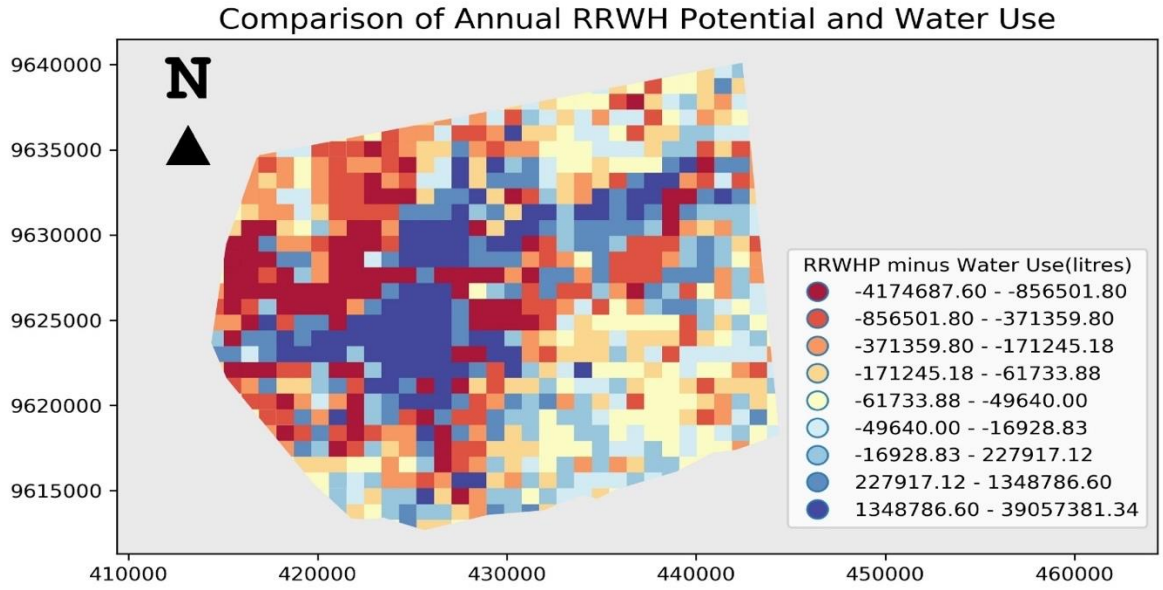


Figure 37. Annual potential satisfaction of water-use by RRWH.

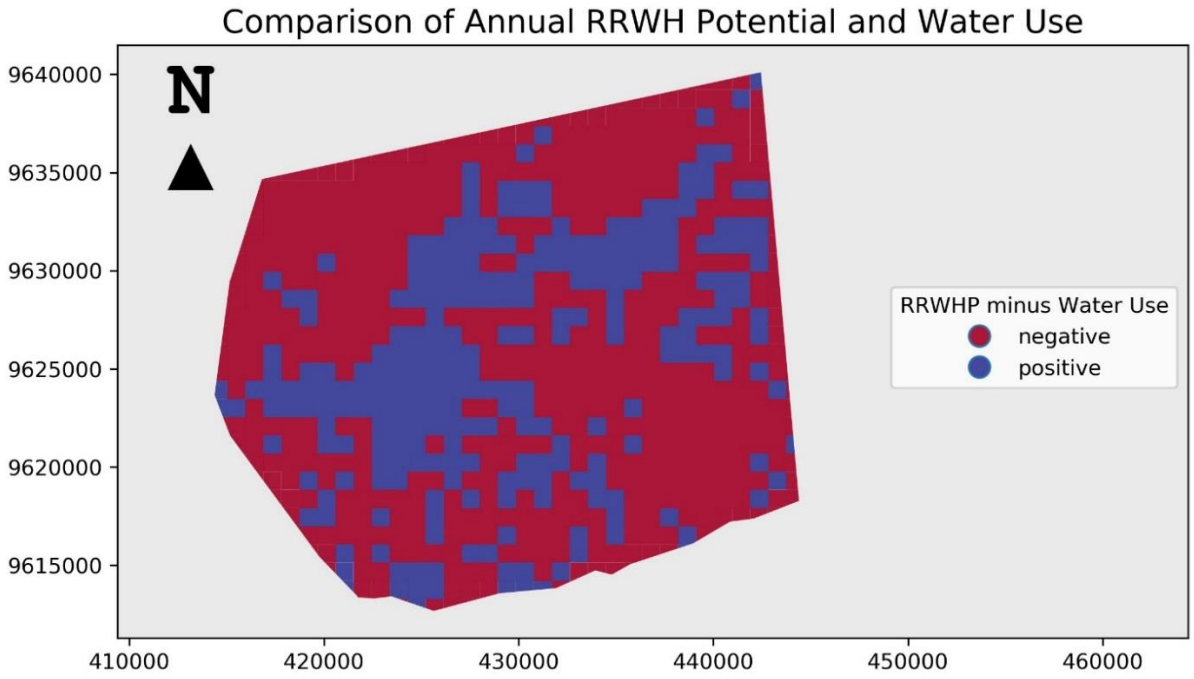


Figure 38. Net satisfaction of water-use by RRWH.

Figure 39 shows that in the dry season from July to August, almost all areas cannot be supported by RRWH, while almost all can be supported during the very rainy period. By and large, about 60 % of the households can be supported by the RRWH system on average (Figure 40).

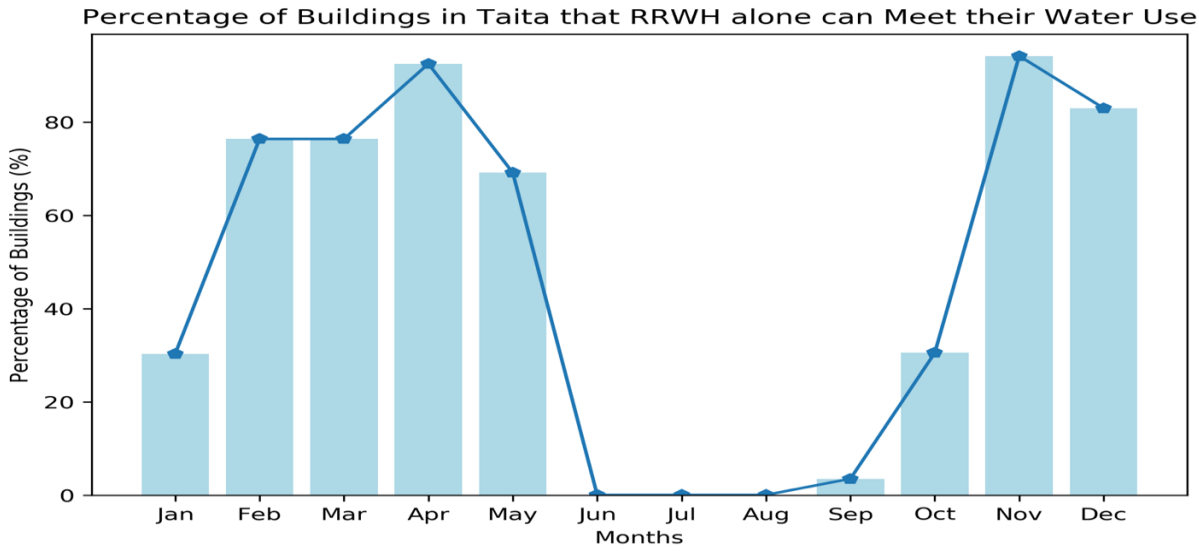


Figure 39. Percentage of households with positive monthly net satisfaction of water use by RRWH.

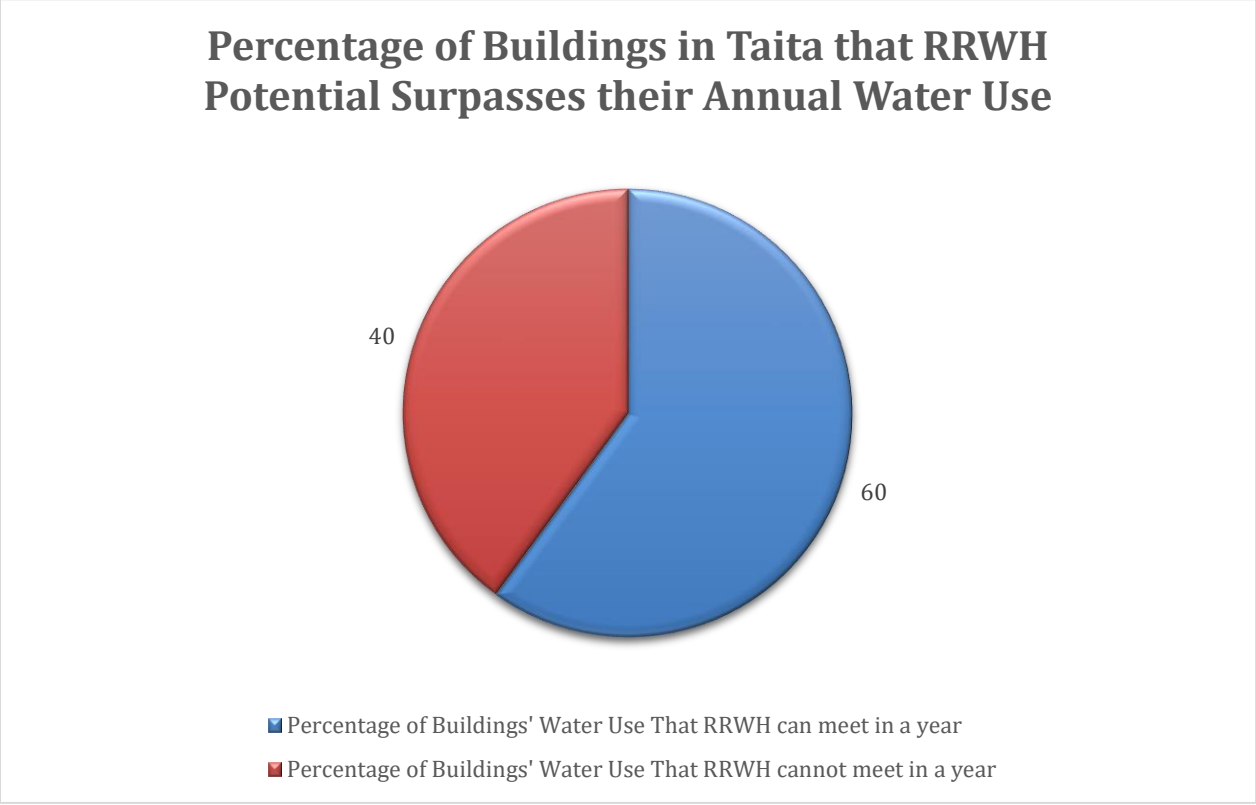


Figure 40. Percentage of households with positive annual net satisfaction of water use by RRWH.

6.5 Household survey

Figure 41 shows that farming, trading and construction work are the most common amongst the Taita people. Also, the average household size is four people (Figure 41).

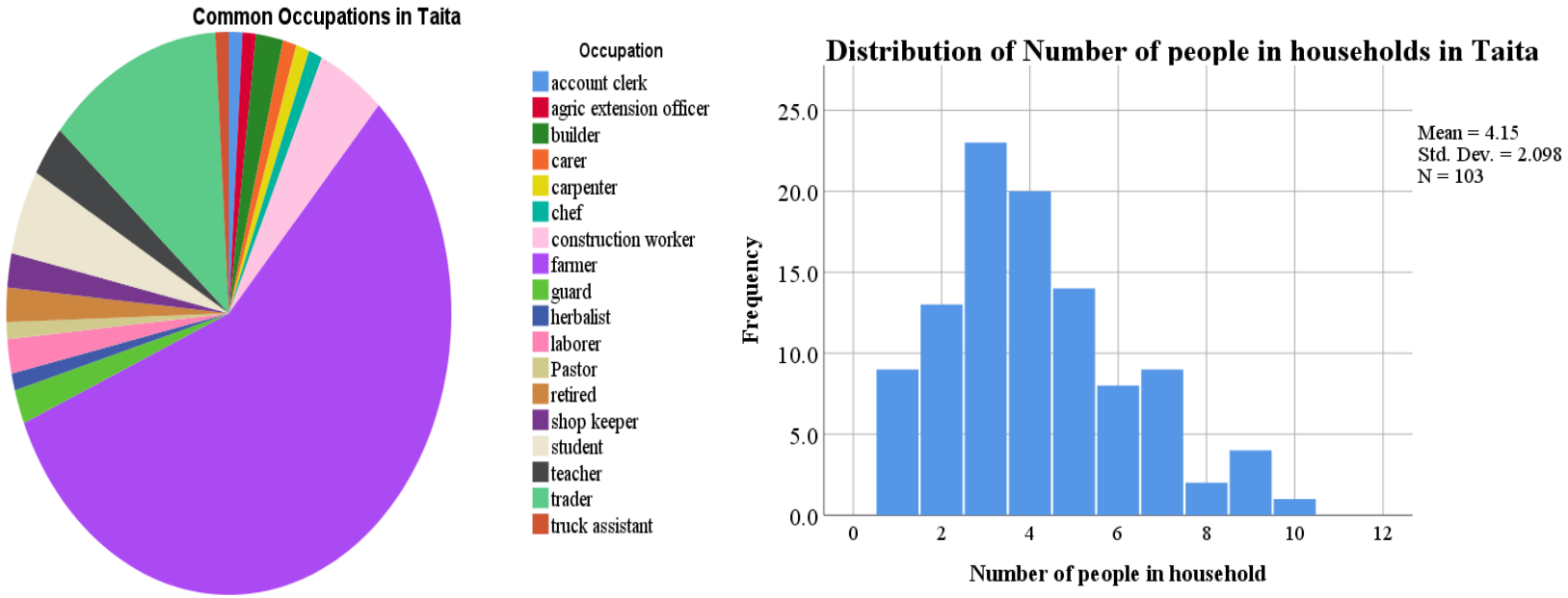


Figure 41. Occupation of respondents on the left and size of the households on the right.

Figure 42 shows that the average daily water consumption in the area is about 136 liters, with little difference of a about four liters between the highland and the lowland. Most of the households use below 400 liters of water per day.

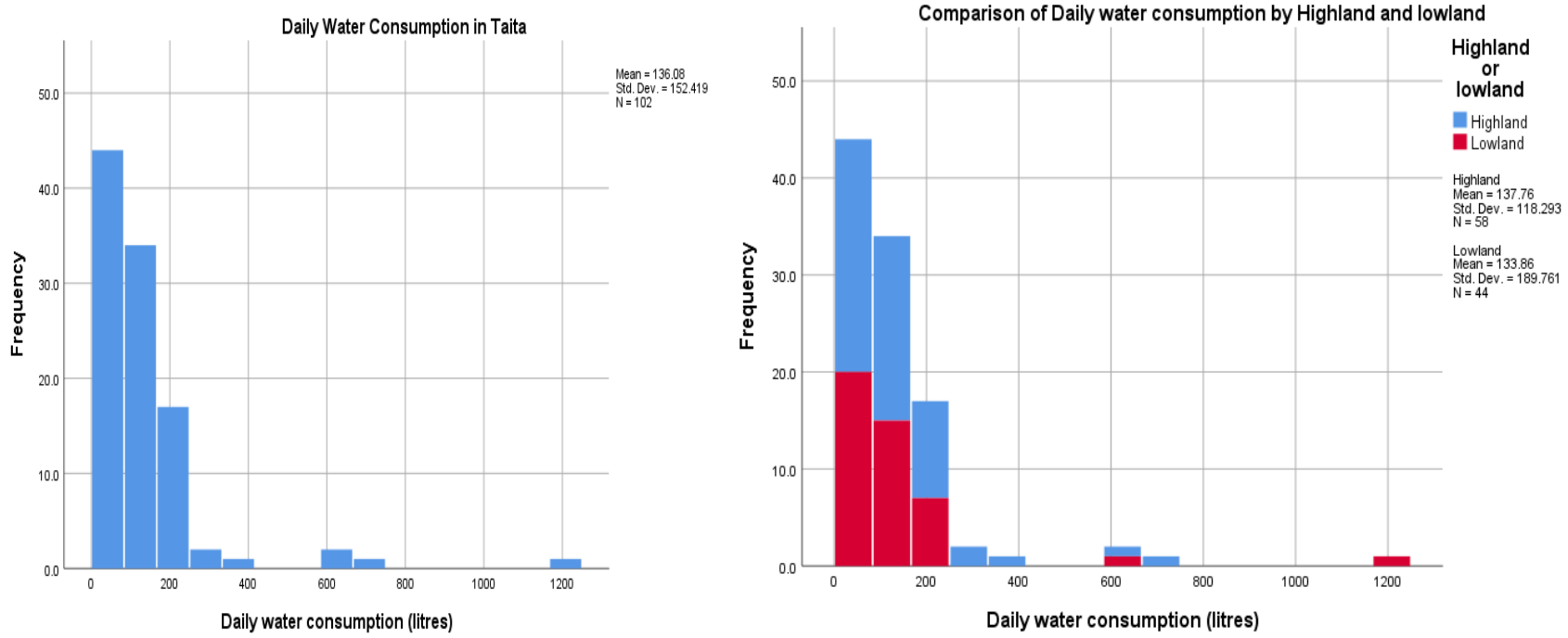


Figure 42 Daily water consumption in the Taita Hills.

On average, people travel for about 43 minutes to get water. Over than an hour is spent traveling to get water in the lowland. It takes about 16 minutes in the highland. Some can also be seen to travel more than 400 minutes to get water (Figure 43).

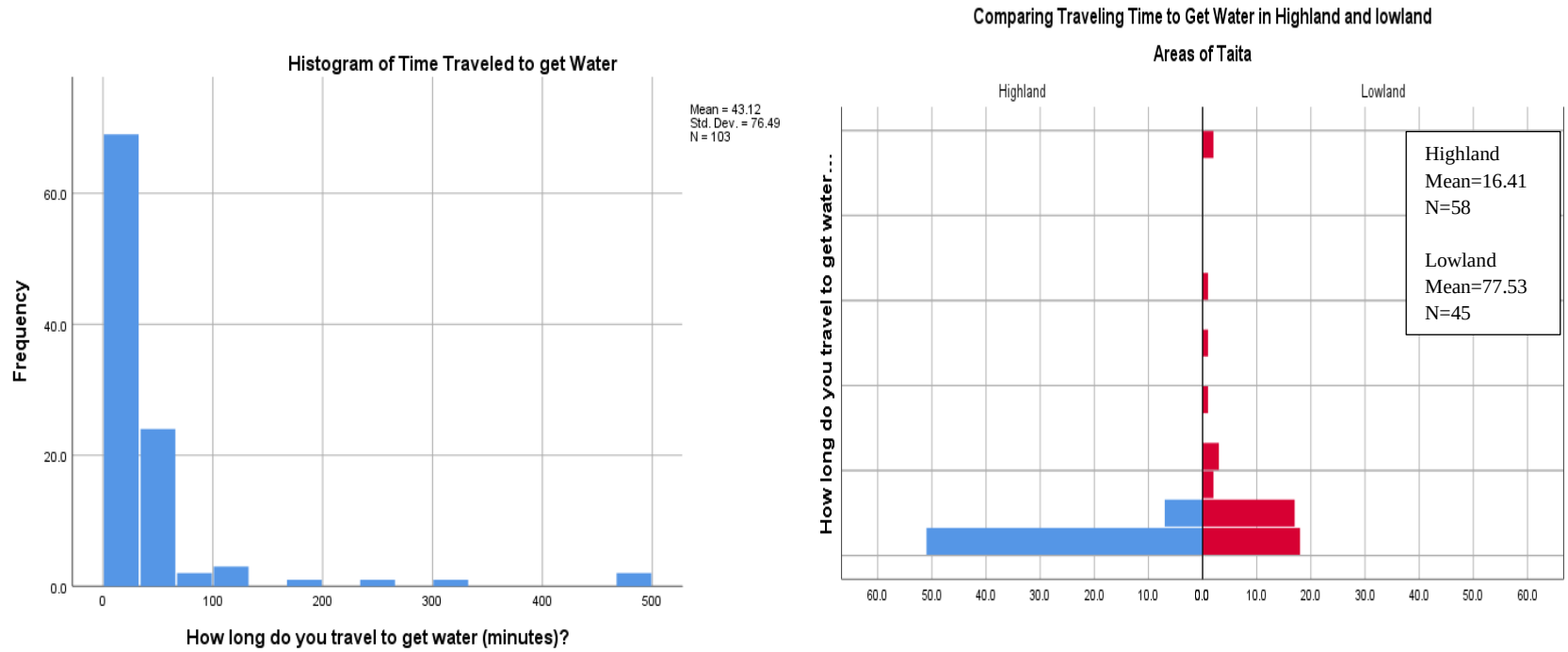
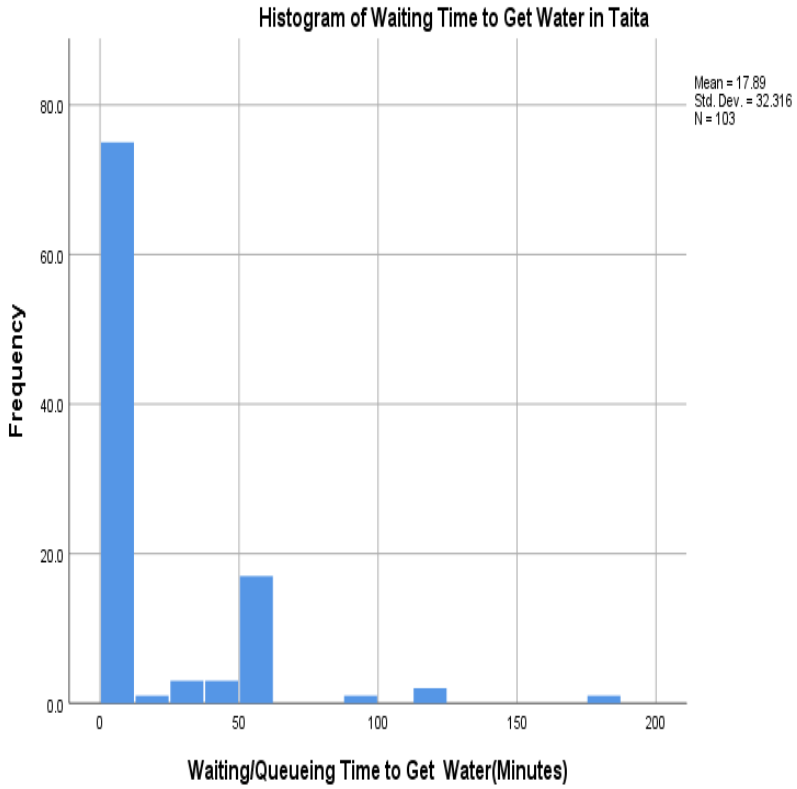


Figure 43. Travel time to get water.

Generally, the average waiting time when getting water in the area is about 18 minutes. People spend about 30 minutes on average in the lowland, waiting or queueing when getting water. In contrast, the waiting time is 8 minutes in the highland (Figure 44).



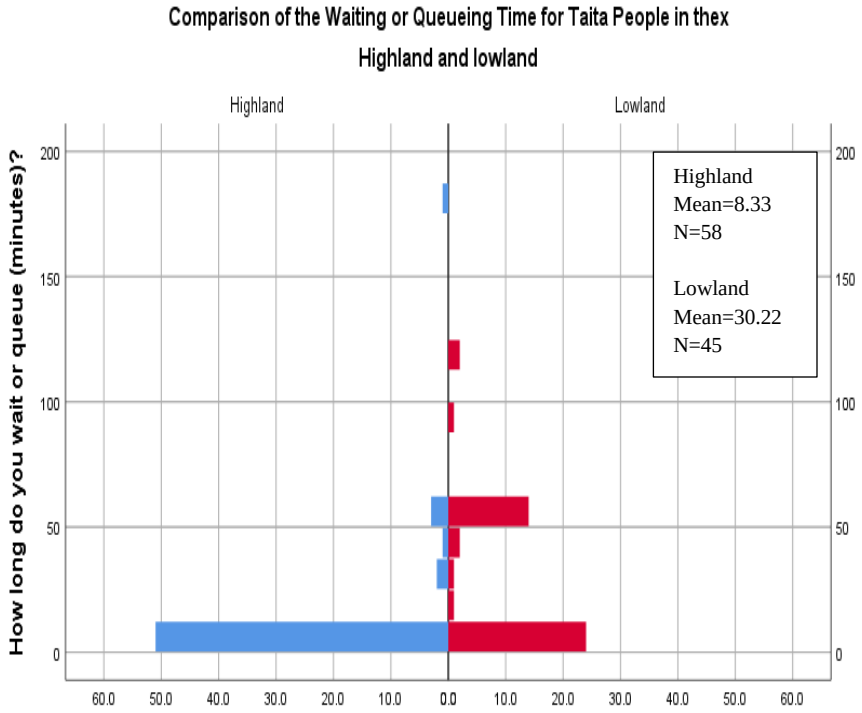


Figure 44. Waiting and queueing time to get water.

More than 50% of the people face water shortage in one form or the other. However, the shortage is experience more in the lowland (Figure 45, Figure 46 and Figure 47).

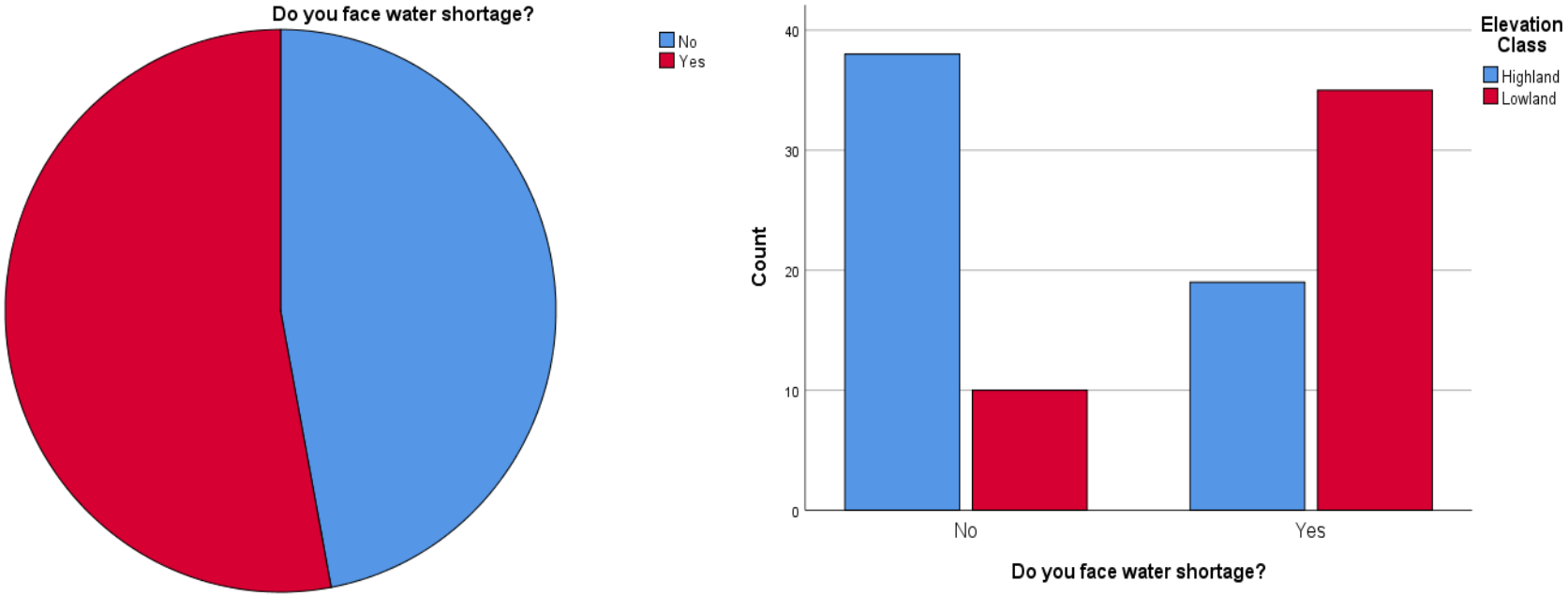


Figure 45. Percentage of households that face water shortage in Taita Hills.

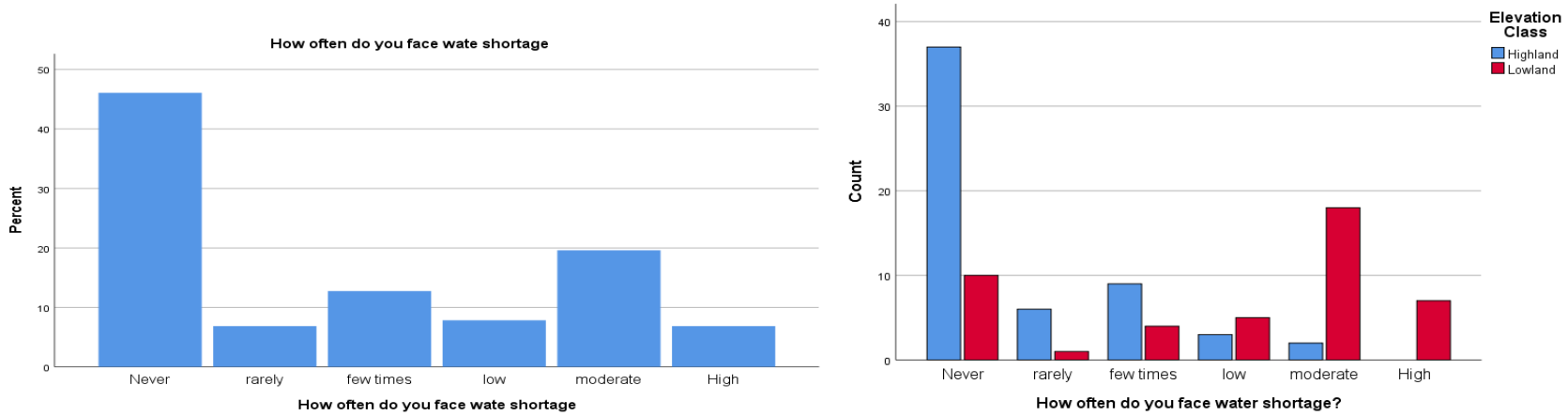


Figure 46. Frequency of water shortage.

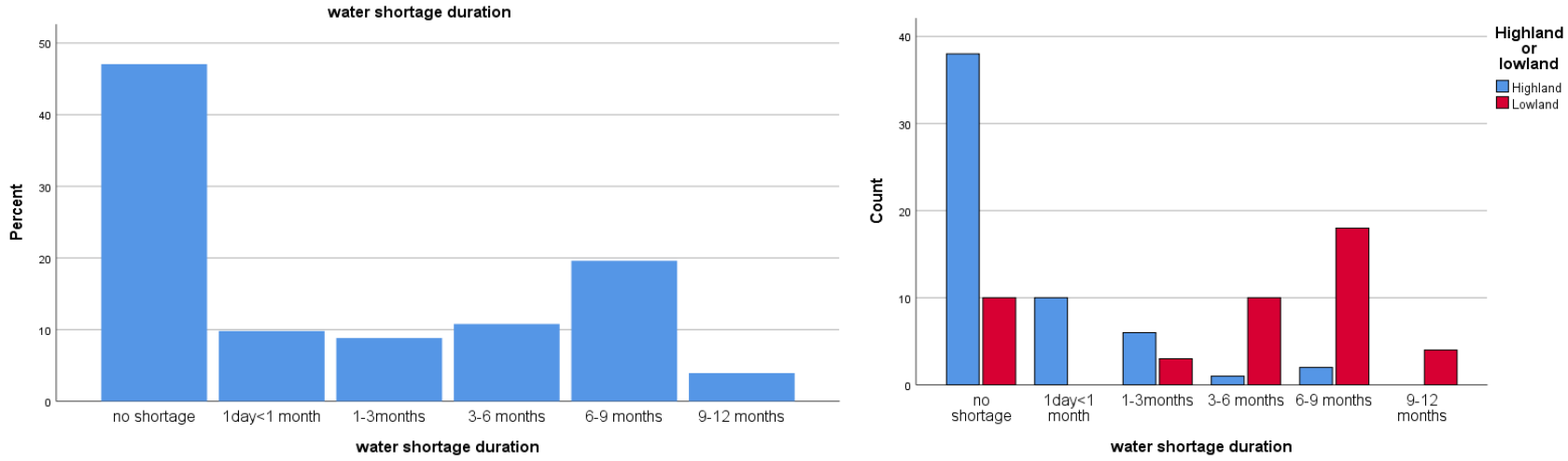


Figure 47. Duration of water shortage.

Figure 48 shows that there are more households with rainwater harvesting system than those without. Most households that do not have the system mostly mentioned that they cannot afford it (Figure 48).

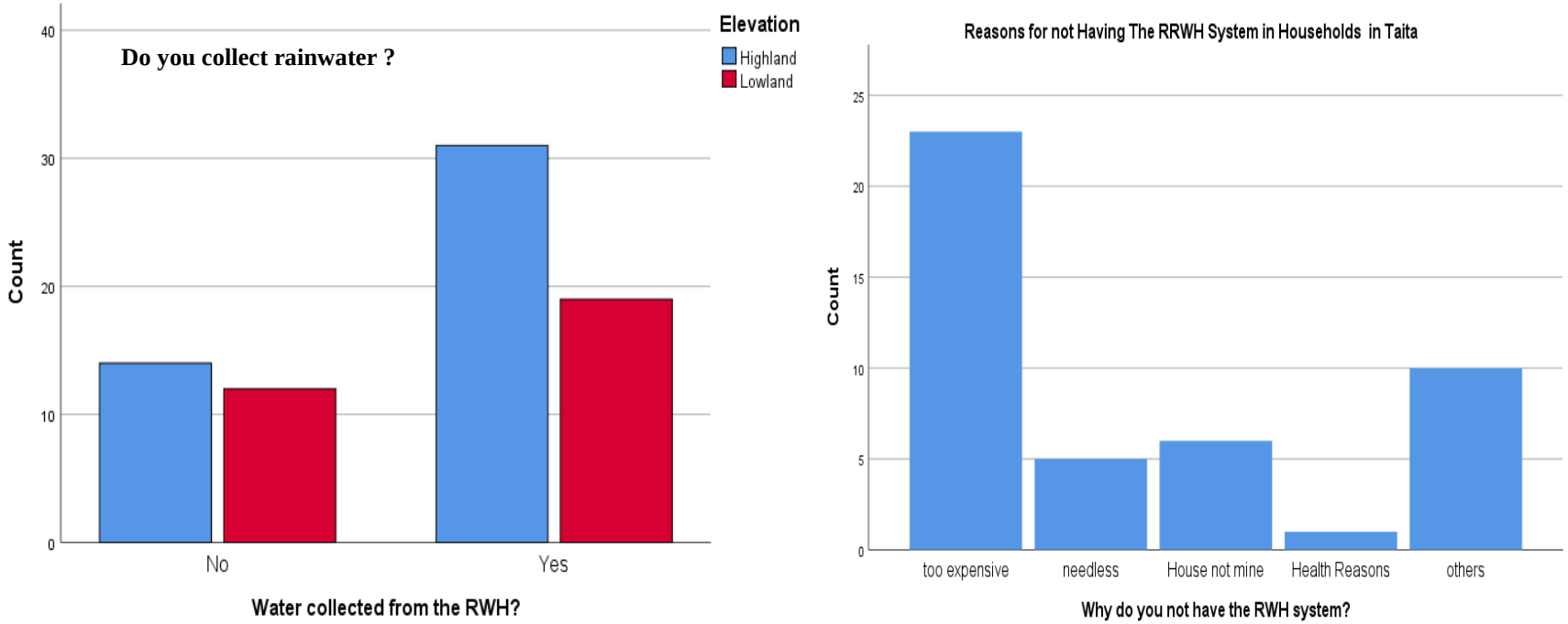


Figure 48. Distribution of households that already use RRWH system on the left, reasons people do not install the RRWH system on the right.

The average tank size used in Taita Hills is about 4,400 liters and most people have less than 20,000 liters tank (Figure 49). On average, the people in lowland spend three times more on water monthly with about 1,500 shillings (15 USD).

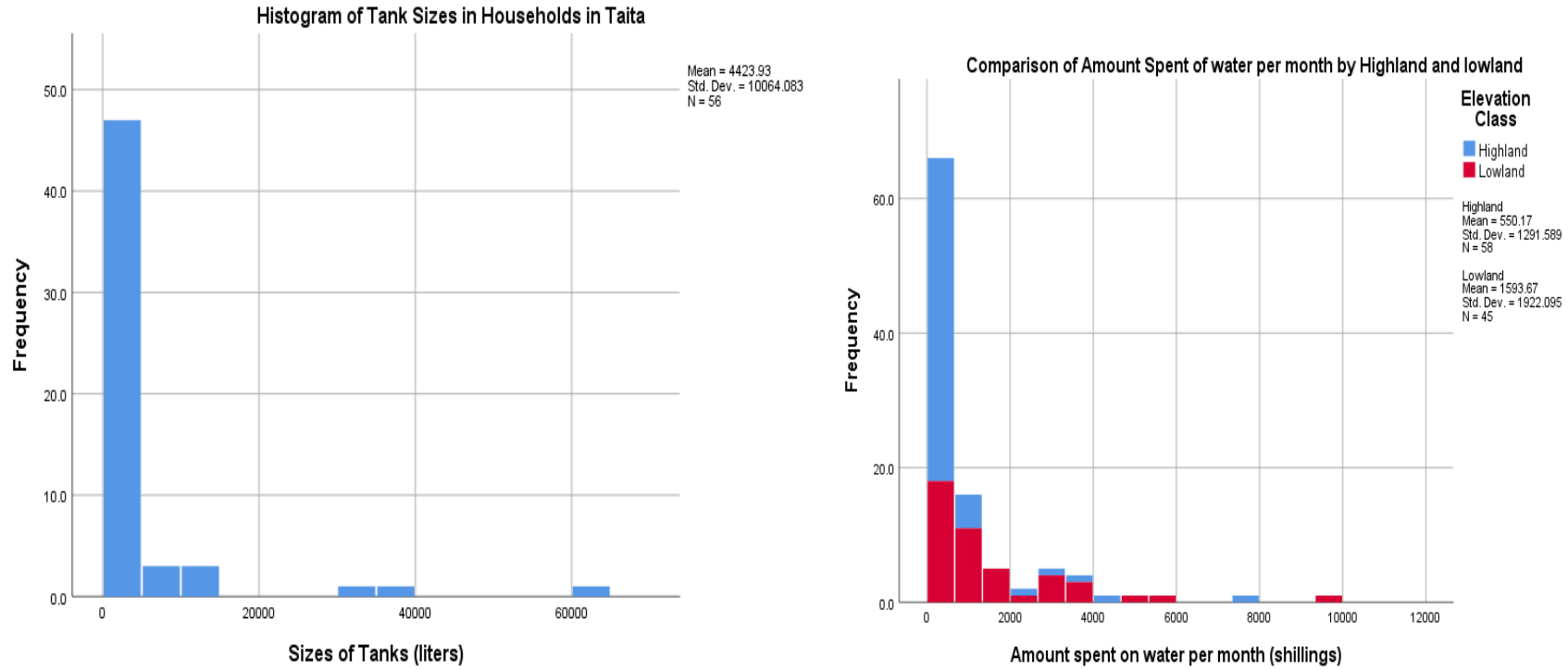


Figure 49. Various tank sizes in different households on the left and amount spent on water monthly on the right.

Generally, people spend on average, about 17,000 shillings (166 USD) to install RRWH system. Also, similar to above, people in the lowland spend on average about 6,000 shillings (147 USD) more than people in the highland (Figure 50).

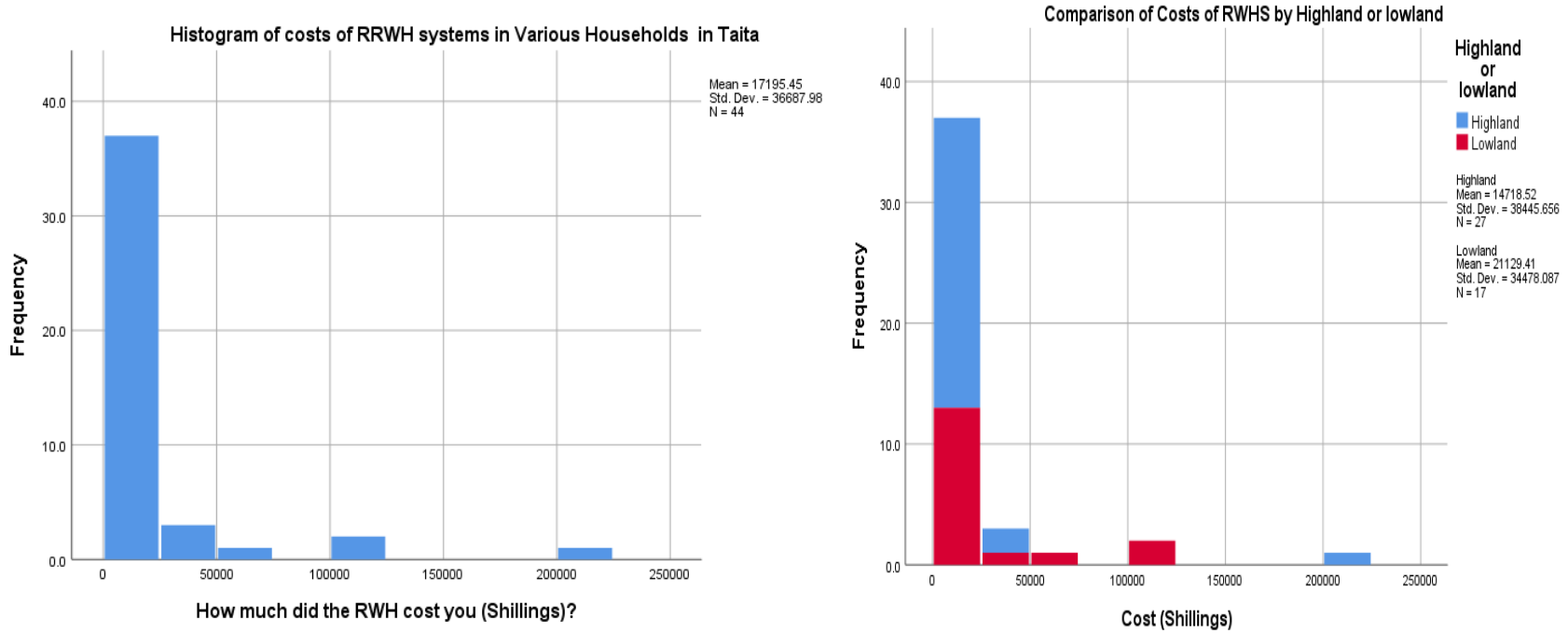


Figure 50. Average cost of installation of rainwater harvesting system in Taita Hills.

Lastly, if provided by the government or organization, the lowland people prefer to have more than twice as big tank size than the highland people (Figure 51).

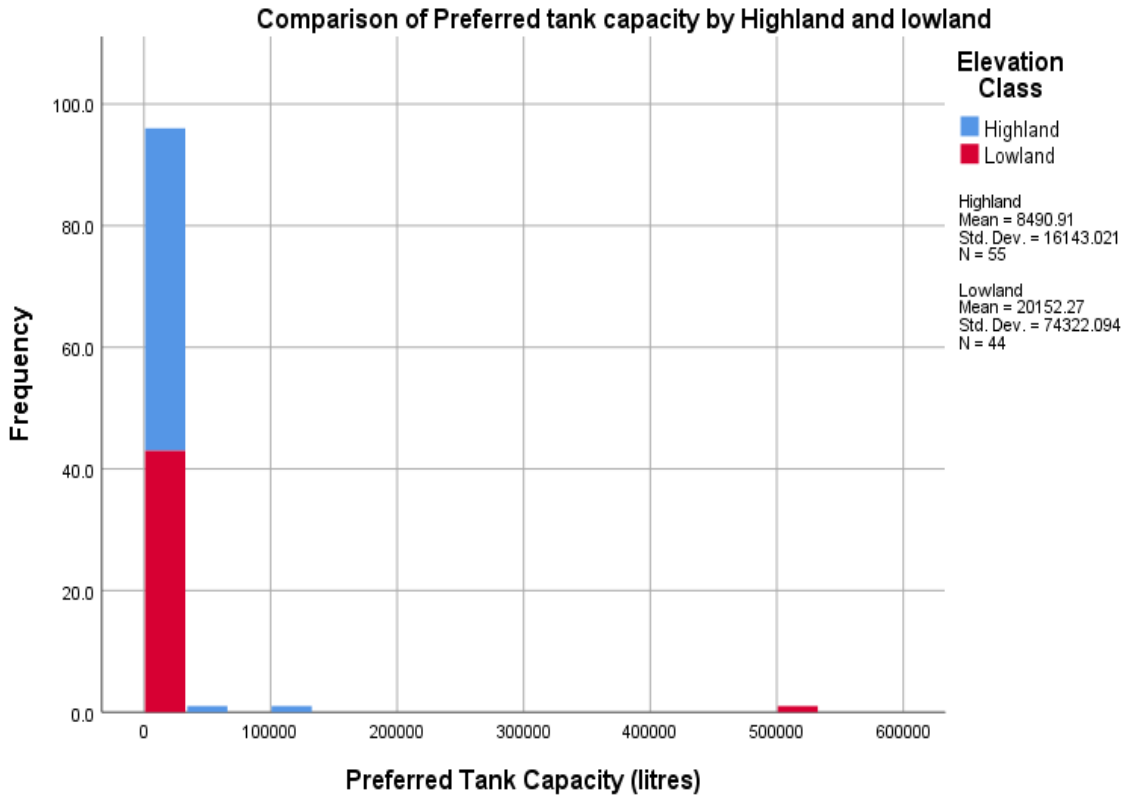


Figure 51. *P*referred tank capacity in highland and lowland area.

7 DISCUSSION

Water scarcity is a global problem and needs to be understood holistically both in local and regional. This way, the appropriate potential solutions can be recognized as there is no panacea. For instance, the same action that solves the water problem in the highland area of Taita Taveta County may not solve that on the lowlands. The focus of this thesis was mapping the potentiality of rooftop rainwater harvesting system and to assess their acceptability by the community. From the foregoing, we have seen that the potential varies across the area and monthly. Utilization of LiDAR data facilitate the coverage of a large area to assess the potentiality in cost-effective way compared to field assessment of roof area. The next I will discuss further about my findings.

7.1 Automatic Building Detection and Lidar

LiDAR data has been used in wide range of studies, as stated earlier. Commonly, many studies utilize already existing footprint for carrying out their modelling (A. Brunn & U. Weidner, 1997; N. Haala, et al, 1998). The problem, however, is that these already extracted buildings are not always available. Such is the case of Taita area. Other studies have also attempted to generate the footprint by deriving edges of image from multispectral reflectance (C. Brenner, 2010). Furthermore, wide range of other methods have been utilized (C. Baillard 1991, S. You 2003; Z. Kim 2004; F. Rottensteiner et al, 2004). In this study, the polygons of the roofs are necessary for the modelling of RRWHP and therefore, had to be generated from the existing LiDAR datasets.

Another interest was to test the impact of data characteristics on house detection. Previous studies have focused more on the algorithms, and very seldom to assess how the point spacing and pulse density of a LiDAR data affects the extraction of building footprints (Goodwin et al., 2009; Shridhar D Jawak, Panditrao, & Alvarinho, 2013; Wang et al., 2007). By comparison of the datasets from different flight campaigns acquired by different sensors and specifications over the same areas, we can find out how point spacing, and pulse density affect the accuracy of building extraction considerably.

The point densities and spacing's are represented in Figure 52 for the Leica data from 2015 with about 3 m pulse density and high point spacing and for Optech data from 2013 with about 10 m

point density. A higher point spacing and lower pulse density means that the data has lower quality and is likely to yield more errors in house detection.

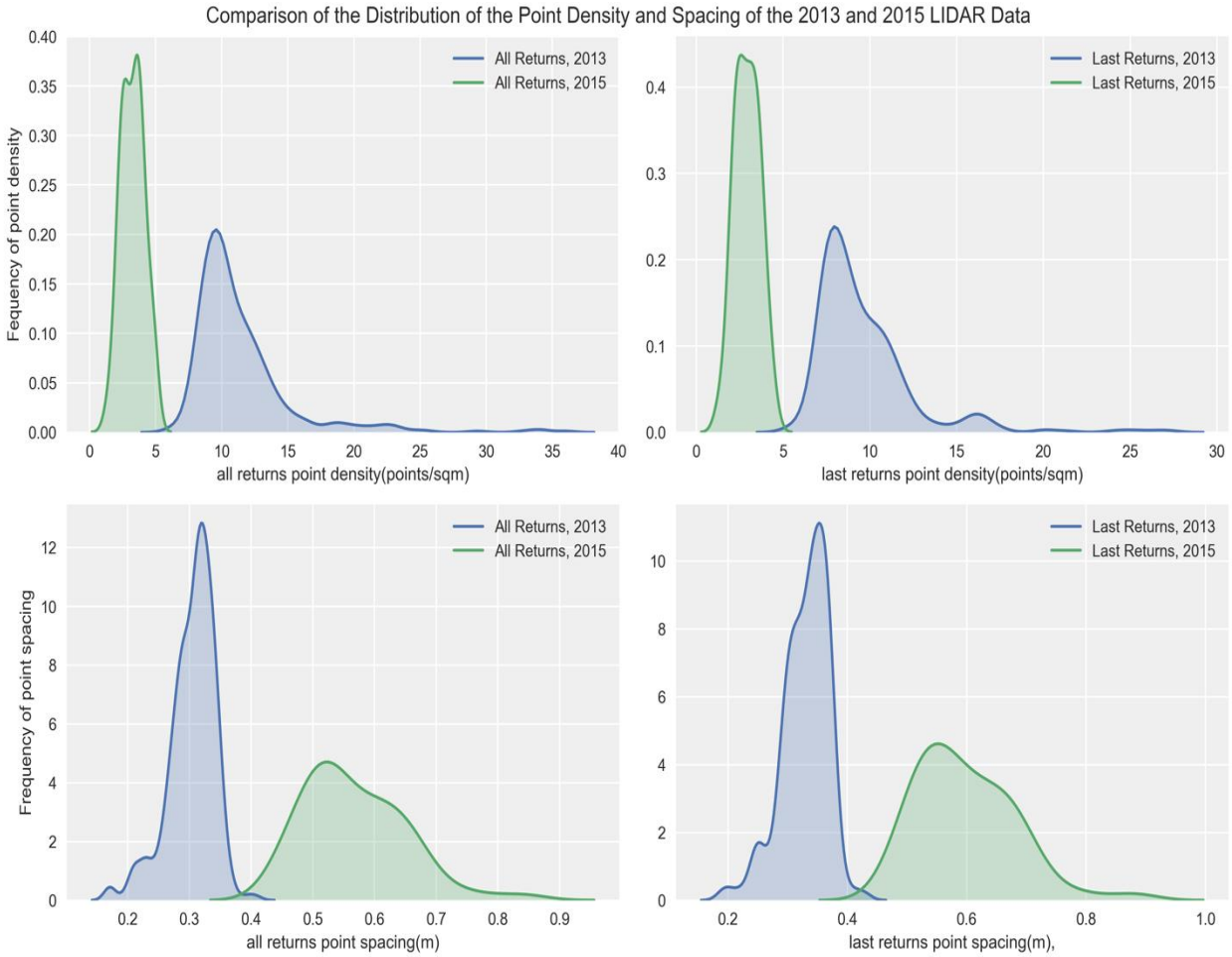


Figure 52. Comparison of the point spacing and pulse density of 2013, 2015 LiDAR data

One of the methods of reducing false positives is the proper definition of removal criteria for polygons' area beyond a threshold. In urban area, $30m^2$ and $35000m^3$ have been used as the lower and upper boundaries for defining the accepted area of buildings (Shridhar D Jawak et al., 2013). Another study in an urban area filtered out areas of buildings less than $10m^2$ (El-Deen Taha & Ibrahim, 2016). However, $10m^2$ and $2500m^2$ were adopted in this thesis based on the investigation of the data and because the houses are rather small in rural area. As a minimum height for the buildings 1.7 m was used compared to 1.5 m used by Shridhar D Jawak et al. (2013) and 3 m by El-Deen Taha & Ibrahim (2016). This was to avoid the inclusion of cars, and other lower objects with smooth surface like car roofs. In the course of the extraction of buildings' footprints, it was shown how much the data quality can also affect the accuracy of feature extraction. For

example, one limitation of LiDAR data is its massive data volume which requires huge computational loads (Hu & Ye, 2013).

7.1.1 Accuracy Assessment and Validation

The pulse density and point spacing were the key factors affecting the accuracy of the extraction. Many studies have focused on the algorithm used (Hu & Ye, 2013; Wang et al., 2007). In this study, however, it was shown how point spacing and pulse density affects the accuracy of the building extraction. Some studies mention the buildings removal criteria, but not the accuracy of the building detection (Shridhar D Jawak et al., 2013). In other studies, different accuracies have been obtained with different data quality (REF).

El-Deen Taha & Ibrahim (2016) used LiDAR data of 10 cm point spacing and pulse density of 65 points/m^2 and extracted buildings with the producer's accuracy of 86% and user's accuracy of 89 %. Goodwin et al. (2009) used 0.7 m point spacing and were able to achieve detection accuracy of 83 % and commission error of 38 %. This study evaluated the accuracy of the polygons' area using aerial photography and field observations, which resulted to an accuracy of 73 % with outliers and 96 % without the two outliers. The highest accuracy utilizing LiDAR data was obtained by Hu & Ye (2007), who detected 200 buildings with 100 % user's accuracy and 94.5 % and 92.1 % producer's accuracies in two test areas, respectively. In similar vein, 80% accuracy was achieved in a study with 380 buildings (Wang et al., 2007), but the study did not make mention the point spacing or pulse density of the data.

The previous studies assess either the accuracy of a dataset in an area or from two different areas. Having analyzed data from same area using two different datasets, I found the Optech data from 2013 to generally reach higher detection rate with user's accuracy of 89 % compared to Leica data of 2015 with 83 % accuracy (Figure 53). The difference is 6 %, but it is still considerable, even though the densities of both data are not so high compared to other studies (El-Deen Taha & Ibrahim, 2016).

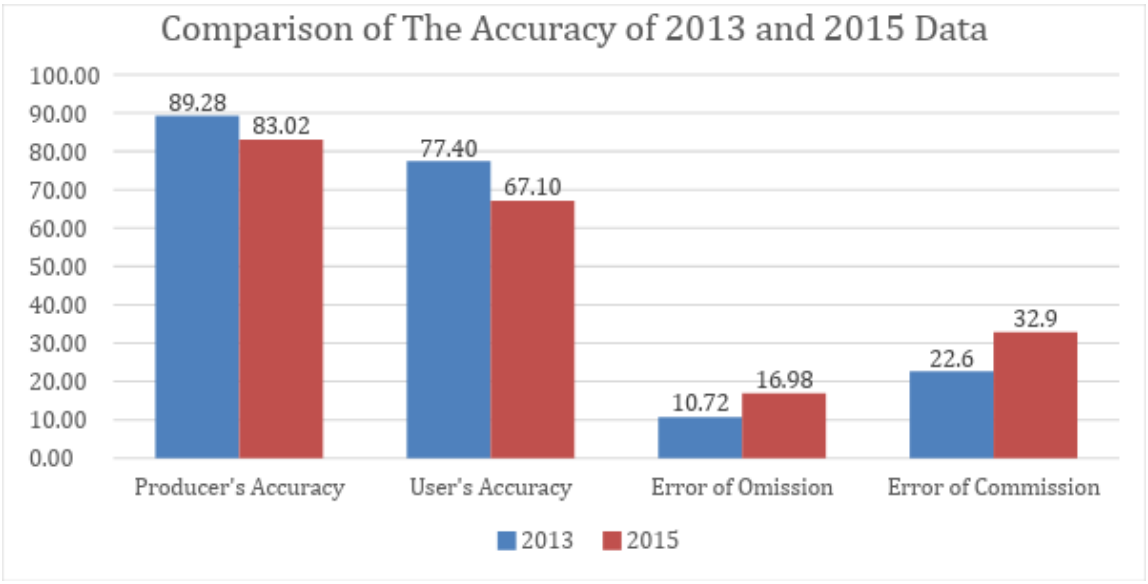


Figure 53. Comparison of accuracies of 2013 and 2015 LiDAR datasets.

Similarly, the user's accuracy of the 2013 Optech data was about 10 % more than that of 2015 Leica data with 67 % accuracy. For 2013 versus 2015, there were about 256 versus 255 cases of one-to-two, 13 versus 17 of one-to-three and 2 versus 4 one-to-four relationships. This is one of the typical errors that occur during automatic extraction, whereby especially buildings in proximity are merged as one polygon. This has also been identified in study by Zeng et al. (2013). The 2015 Leica data has lower pulse density and higher spacing compared to data from 2013. The combined effects are: lesser quality, and as a consequence more errors are bound to occur frequently (Figure 54).

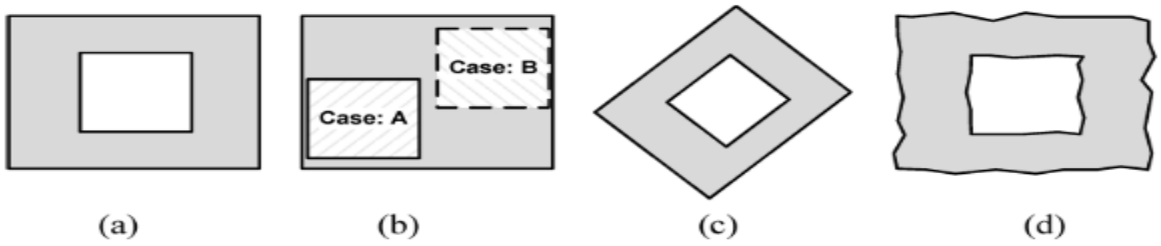


Fig. 2. Hypothetical examples of different building extraction results compared with the reference. (a) Reference building boundary. (b) Extracted result 1: an extracted building with the courtyard displaced to A or B. (c) Extracted result 2: a rotated building. (d) Extracted result 3: a polygon with irregular borders.

Figure 54. Illustration of common issues with automatically extracted polygons (Zeng et al., 2013).

Evidently with lower point spacing and higher pulse density, a better result could have been achieved in this study. The accuracy assessment was done for building detection and not for estimation of the roof area due limited resources in the field work. To conclude, the 2013 LiDAR data is of better quality and is more suitable for building extraction, but covers only 10 by 10 km, while 2015 data covers more than 1000 km2.

7.2 Spatiotemporal Assessment of Rooftop Rainwater Harvesting Potential

Besides the building footprints, rainfall data is another important data when evaluating RRWHP. The use of rainfall data from various weather stations managed by the Taita Research Station was assessed after aggregating and inspecting to be insufficient for modelling rainfall for each month for the whole study area. CHELSA rainfall model was used and validated with the data from the weather stations managed by Taita Research Station. The model shows that the rainfall pattern in Taita is governed by topography and elevation. It rains more in the highlands, while lowlands are drier. June, July and August also appear to be the driest months. The influence of the ITCZ can also be seen in the monthly rainfall creating a bimodal pattern, with April and November having the highest rainfall. The CHELSA rainfall model shows quite good accuracy of 72 % and RMSE of 38.74 mm when assessed with data from Taita Research Station weather stations.

The coefficient factor was estimated to be 70 % even though the galvanized iron sheets largely used in the area are considered to have a coefficient of 85 % to 90 %. I decided not to overestimate the rainwater harvesting potentiality as it is a sensitive issue that affects the lives of many people. In addition, many roofs are not in a good condition and are likely to have lower coefficient as pointed out by Zhe Li & Fergal Boyle (2010). In similar vein, an image classification could have been done for the roof classes, but it may have introduced more errors than into the analysis. In addition, the field survey showed that only a few houses had other roof types besides galvanized iron sheet. In fact, all the households visited had all the galvanized iron sheets, which is a popular choice because it is affordable, accessible, and is good for water harvesting. The first flush is considered to reduce RRWHP, but as this water is useful for other domestic purposes such as irrigation of gardens or flushing of toilet, the first flush was also included to RRWHP.

One of the key point is that it is not enough to focus on the spatial aspects of rooftop rainwater harvesting because it only yields rough results that might not be compelling for stakeholders to take strategic actions. The general pattern of rainwater harvesting potentiality based on the input multi-criteria can be seen in the result maps, but they do not tell what the real socio-economic situation in the area is. Some households do not face water shortage while some others face a long-term shortage. In the other words, one may wonder why people utilizing rainwater harvesting system in the highlands still suffer water scarcity. An annual precipitation map does not show that all the areas in Taita Hills do not have considerable water potential during the dry seasons and especially in June, July and August. Furthermore, by integrating the water consumption data derived from the field, it was seen that even though an area has a high-water potential, it may not cater fully for the needs of households in the area taking into account the daily water consumption and the inadequacy of proper water storage tanks. Generally, water should be stored for the dry months, as the RRWHP is lower than the consumption for almost the entire area in the dry season.

7.3 Socioeconomic Context and Potential Impact of Rooftop Rainwater Harvesting System on the Taita people

The average number of people in households in Taita is four, while some households have as much as ten people. The most common occupations are farming, trading and construction work. Generally, the average water consumption per day is about the same in the highland and in the lowland with 138 liters and 134 liters, respectively. However, the people in the lowlands are faced with more challenges in retrieving water. Even though the water consumption is about the same, people in the lowland spend about 1,600 shillings (16 USD) monthly on water. This is three times more than people in the highland spendOn the other hand, in the highlands the majority do not face water shortage. In the same vein, the lowland people face shortage more frequently over six months compared to people in highlands who face water shortage for less than three months.

The overall average time it takes to get water is 43 minutes. This does not give a detailed picture of the situation, as the people in the highland spend on average 16 minutes to get water while people in lowland spend 78 minutes. It is even worse in some cases. For example, two of

the respondents spent about 6 to 8 hours daily to get water. These are women who could have used this time to do something productive and even educate themselves to improve their living standard. To get a deeper perspective, people in the lowland spend about four times longer waiting or queueing for water compared to their highland counterpart. On average, people wait for about 8 minutes at the water point when getting water.

It was found out that most people already collect rainwater. The average tank size is about 4,400 liters. Also, the average cost of installing the rainwater system is about 17,000 shillings. However, people in the lowland spend about 21,000 shillings, which is about 6,000 shillings more than in highlands. The survey also revealed that the lowland people feel more need to store water. When asking what tank capacity, they would prefer, the people in the lowland want to have on average 20,000 liters tank, while in the highland 8,500 liters tank was sufficient.

The most common water storages are plastic or concrete tanks, which are more durable than metal tanks. Many concrete tanks have been designed and constructed by the Danish International Development Agency (DANIDA) in late 1990s. Major challenges are caused by air quality and roof cleanliness in Taita Hills. Especially in the lowlands, it is very dusty causing poor air quality and contaminated water harvested from the roofs. Nevertheless, a few households have devised means to purify the water with chemicals or even solar energy (Figure 55). Therefore, even the first flush water quality in Taita can be improved.

Figure 55. Chemicals or solar energy used for purifying harvested water.

Many roofs were built for enabling water harvesting, for example they were built steeper for water harvesting than needed. The material used is mostly galvanized steel, with the semi-circular gutter being the most effective for transporting rainwater to tanks. The enclosed circular storage tanks are the most common in the area placed on the ground or below the ground. Most typical are the above-ground tanks located close to the homes. All in all, many advantages of the aboveground tanks outweigh the underground tanks, for example above-ground tank is cheaper to install and easier to maintain.

7.4 Sustainability Issues

Sustainability is a paramount consideration to bear in mind when embarking on a project such as the domestic rooftop rainwater harvesting. It is the design and building of a structure or project that is properly maintained financially, technically and environmentally (Joshua, 2008). This is a herculean task that requires the efforts of both the people and the government of Taita Taveta County. Problem of maintenance is a major issue plaguing many projects (Lindqvist, 2005). Despite the rise in the use of RWH system in Kenya, many of those have been short lived (Critchley, 1991a). It is utmost critical to ensure that when embarking a mass provision of RRWH system, measures are put in place to sustain them and make sure that they actually solve the people's water problem. The interest in and knowledge of rainwater harvesting has been increasing during recent decades within private sectors, non-governmental, governmental and international organizations due to the success in mitigating water scarcity, especially in rural areas. Many Kenyans face water shortage during the dry seasons and are very willing and open to utilizing RRWH system to help them store the water for the dry months, as is the case of the people in the lowlands of Taita Taveta County.

7.4.1 Project Feasibility

A feasibility study is crucial to understand the past projects and the status quo, from an economic, technical, social and environmental perspective. With this, proper plans can be made in building more sustainable rooftop rainwater harvesting projects. There should be a synergy

between all the stakeholders including individual households, the community at large, local institutions and the government (Joshua Black et al., 2012).

It is necessary to have a cost and benefit analysis for proper evaluation of the feasibility of the RRWH project. The goals of the project need to be set beforehand and usually, the benefits are considered from economic perspective i.e. profit making. However, special considerations must be put in place for projects targeting the poorest of the populace. The benefits have to be considered as not primarily for profit making but for improving livelihoods and wellbeing of the people (Joshua Black et al., 2012).

A realistic loan scheme could be set up for the project in order to ameliorate water scarcity in many households. The income derived from the loan should also be reinvested into maintaining the project. The water use can be maximized by directing the overflow from tank into gardens for irrigation. Another recommendation would be to excavate ponds for, say, every 10 houses, where excess rainwater can be diverted into and can be utilized during dry spell. It is salient to note that the water needs in cooler highland areas such as Taita Hills, are often neglected by the government, because of the misconception that their water needs are met (Joshua Black et al., 2012). However, as evidenced in this study, these communities also have months without rainfall, hence, the water shortage. There is a need for proper evaluation of the situation obtainable in various locations. Addressing the water issue in every location, and not just in the driest areas, can improve livelihoods and help every household to move towards self-sufficiency.

Furthermore, certain questions have to be answered when starting a RRWH project. It is important to understand who will be implementing the projects? The government or NGOs or individual households? What is the demographic situation in the region? What kind of households are we targeting? What is the existing situation with water resources and roof water harvesting? Can RRWH, at least, considerably ameliorate water scarcity and poverty? How would the beneficiaries be decided? Pinpointing such issues can help to prevent nepotism, favoritism and corruption (Joshua Black et al., 2012). It is also crucial to make realistic goals and promises, as this requires the trust of the community on the project. Similarly, the political situation of the community must be considered when allocating rainwater harvesting system. This can help to

understand the likelihood of corruption that might exist. Nevertheless, it is recommended to incorporate local community elders and organizations in the process (Brown, 2008).

Another point of consideration is the lifecycle of the project. Many projects begin to degenerate soon after the funding stops. Therefore, the project has to be executed strategically, to have a long lasting RRWH system (Mati et al. 2008, Nijhof & Shrestha 2010). As with many other regions of sub-Saharan Africa, women are known to bear the brunt of the water scarcity (Ngigi, 2009). Many spend substantial amount of time during the day just to get water. This is time that could have otherwise been put into other productive activities such as education. An example is two of my female respondents in their early 20s whom travel 6 hours daily to get water. Hence, it is important to involve women when deciding on the allocation and management of the system (Worm & Hattum, 2006). The improvement of the domestic rainwater system will positively affect women and improve gender equality and fairness, since women can be more productive if not using their time for water. RRWH system cannot solve all the problems, but it will go a long way to empower the people, as they become more self-sufficient. This will yield better productivity, as people can channel the energy and time spent on getting water to do other productive things. Consequently, the quality of lives of the people will increase considerably.

8 CONCLUSIONS

It is important to have LiDAR data with high pulse density and low point spacing for automatic building detection and extraction. Also, future studies should include the validation of roof areas on a substantial scale. Water harvesting potential varies spatially and temporally, hence the temporal aspect should not be neglected in following studies. Also, focus should not be on the temporal pattern alone, as that may be too broad generalization. It can be best understood spatiotemporally, which can provide better framework for decision making when allocating RRWH system in an area.

While the lowland areas of Taita Taveta County have lower RRWHP than highland areas, they both face water problems and also require water intervention projects. Excess water should be stored strategically towards the time when RRWHP is low. RRWH system would be a viable solution but has to be supplemented with more allocations to the lowland. The potential impacts of RRWH on the Taita people would be significant and would include increased access to necessity of life – water, reduced water problems, better quality of life, health benefits, improved gender equality, increased productivity, more time for education, improved hygiene, and less dependence on the government.

9 ACKNOWLEDGEMENTS

I am thankful for the Taita Research Station of the University of Helsinki for facilitating my field work, and for Taita Research Station Fund of the Alumni Association of the University of Helsinki to fund it in August 2018. I was getting help for the field work planning by Dr. Tino Johansson and Mr. Anti Autio and valuable assistance from Mr. Muhia Gicheru, Darius Kimuzi and Mwadime Mjomba of the research station. I am grateful for the supervision from Dr. Mika Siljander and Prof. Petri Pellikka and for the idea for the thesis by Dr. Janne Heiskanen. Also, my heartfelt gratitude goes to all my other teachers and colleagues. You have all made a huge positive impact on me.

The outcome of this project is part of TAITAWATER project (Integrated land cover-climate-ecosystem process study for water management in East African highlands) and SMARTLAND project (Environmental sensing of ecosystem services for developing climate smart landscape framework to improve food security in East Africa) funded by the Academy of Finland, and CHIESA project (Climate change impacts on ecosystem services and food security in Eastern Africa - increasing knowledge, building capacity and developing adaptation strategies) funded by Ministry for Foreign Affairs of Finland.

REFERENCES

Abdulla, F. A. (2009). Roof rainwater harvesting systems for household water supply in Jordan. *DES*, *243*(1–3), 195–207. https://doi.org/10.1016/j.desal.2008.05.013

Bartram, J., & Howard, G. (2003). Domestic Water Quantity , Service Level and Health. *World Health Organization*, 39. https://doi.org/10.1128/JB.187.23.8156

Bourke, Paul (July 1997). "Calculating the area and centroid of a polygon"

Critchley, W. (1989). Building on a tradition of rainwater harvesting. *Appropriate Technology*. https://doi.org/10.1016/S0022-3476(96)70434-1

Dadhich, G., & Mathur, P. (2016). A GIS based Analysis for Rooftop Rain Water Harvesting. *International Journal of Computer Science & Engineering Technology (IJCSET)*, *7*(04), 129–143.

De Smith, Goodchild & Longley: Geospatial analysis – a comprehensive Guide to principles techniques and software tools

Dowman, I. J. (2004). Integration of LIDAR and IFSAR for Mapping. *International Archives of Photogrammetry and Remote Sensing*, *35*(September), 90–100. https://doi.org/10.1016/B978-1-42246-332-1.50016-6

El-Deen Taha, L. G., & Ibrahim, R. E. (2016). New object based model for automatic building extraction by integrating LiDAR point clouds and LiDAR derived layers. *International Journal of Circuits, Systems and Signal Processing*, *10*, 306–315.

Erdogan EH, Pellikka PK, Clark B. 2011. Impact of land cover change on soil loss in the Taita Hills, Kenya between 1987 and 2003. Int J Remote Sensing. 32:5919–5945.

Gaikwad, S. D. (2015). Application of Remote Sensing and GIS in Rainwater Harvesting: A Case from Goa, India, *6*(1), 633–639.

Giridhar, M. V. S. S., S, C. B. A., & Viswanadh, G. K. (2013). Identification of suitable locations for rooftop rainwater harvesting structures, *2*(2), 101–106.

Goodwin, N. R., Coops, N. C., Tooke, T. R., Christen, A., & Voogt, J. A. (2009). Characterizing urban surface cover and structure with airborne lidar technology. *Canadian Journal of Remote Sensing*, *35*(3), 297–309. https://doi.org/10.5589/m09-015

Gould, J. (1999). Contributions relating to rainwater harvesting. *Contribution to the World Commission on Dams, Thematic Review IV*. https://doi.org/10.1038/nchem.744

Gould, J. (2015). Rainwater harvesting for domestic supply. In *Rainwater Harvesting for*

Agriculture and Water Supply. https://doi.org/10.1007/978-981-287-964-6_8

Helmreich, B., & Horn, H. (2009). Opportunities in rainwater harvesting. *Desalination*. https://doi.org/10.1016/j.desal.2008.05.046

Hermunen, T., A. Keskinen, M. Lanne, K. Masalin & T. Sirviö (2004). Mwatate – trading post with soil erosion problem. In Pellikka, P., J. Ylhäisi & B. Clark (eds.). Taita Hills and Kenya, 2004 – seminar, reports and journal of a field excursion to Kenya. Expedition reports of the Department of Geography, University of Helsinki 40, 96– 101. Also available in http://www.helsinki.fi/science/taita/publications.htm.

Hu, X., & Ye, L. (2013). A FAST and SIMPLE METHOD of BUILDING DETECTION from LIDAR DATA BASED on SCAN LINE ANALYSIS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *2*(3W1), 7–13. https://doi.org/10.5194/isprsannals-II-3-W1-7-2013

ICRAF, & UNEP. (2005). Potentail for Rainwater Harvesting in Africa: A GIS Overview. *October*, *I*(October), 27.

Jaetzold, R. & H. Schmidt (1983). Farm management handbook of Kenya. Natural conditions and farm management information. VOL. 2, part C. 411 p. Ministry of Agriculture, Kenya.

Jawak, S. D., Panditrao, S. N., & Alvarinho, L. (2013). Validation of High-Density Airborne LiDAR-Based Feature Extraction Using Very High Resolution Optical Remote Sensing Data. *Advances in Remote Sensing*, *2013*(December), 297–311. https://doi.org/10.4236/ars.2013.24033

Jawak, S. D., Panditrao, S. N., Luis, A. J., & Sada, H. (2014). AIRBORNE LIDAR AND HIGH RESOLUTION SATELLITE DATA FOR RAPID 3D FEATURE EXTRACTION, *XL*(December), 9–12. https://doi.org/10.5194/isprsarchives-XL-8-573-2014

Karger, D.N., Conrad, O., Böhner, J., Kawohl, T., Kreft, H., Soria-Auza, R.W., Zimmermann, N.E., Linder, H.P. & Kessler, M. (2017) Climatologies at high resolution for the earth's land surface areas. Scientific Data 4, 170122.

Karger, D.N., Conrad, O., Böhner, J., Kawohl, T., Kreft, H., Soria-Auza, R.W., Zimmermann, N.E., Linder, H.P., Kessler, M. (2017) Data from: Climatologies at high resolution for the earth's land surface areas. Dryad Digital Repository. https://doi.org/10.5061/dryad.kd1d4

Krhoda, G.O. (1998). Conflicts in resource utilization resulting from highland-lowland interactions: A study of Taita Hills, Kenya. In Ojany, F.F. (ed). African mountains and highlands. Planning for sustainable use of mountain resources, 25-39. The United Nations University, Tokyo, Japan.

Liaw, C. H., & Chiang, Y. C. (2014). Framework for assessing the rainwater harvesting potential of residential buildings at a national level as an alternative water resource for domestic water supply in Taiwan. *Water (Switzerland)*, *6*(10), 3224–3246. https://doi.org/10.3390/w6103224

Lupia, F., Baiocchi, V., Lelo, K., & Pulighe, G. (2017). Exploring Rooftop Rainwater Harvesting Potential for Food Production in Urban Areas. *Agriculture*, *7*(6), 46. https://doi.org/10.3390/agriculture7060046

Malesu, M. M., & Resilience, B. (2014). *Rainwater Harvesting Inventory of Kenya . An overview of techniques , sustainability factor and stakeholders* . https://doi.org/10.13140/2.1.2299.5208

Martinson, D. B., & Lucey, A. D. (2004). Reduction of Mixing in Jet-Fed Water Storage Tanks. *Journal of Hydraulic Engineering*, *130*(1), 75–81. https://doi.org/10.1061/(ASCE)0733-9429(2004)130:1(75)

Mati, B., Bock, T. De, Malesu, M., Khaka, E., Oduor, A., Nyabenge, M., & Oduor, V. (2007). *Mapping the Potential of Rainwater Harvesting Technologies in Africa A GIS overview on development domains for Mapping the Potential of Rainwater Harvesting Technologies in Africa : A GIS overview and atlas of development domains for the continent and ten*.

Mwenge Kahinda, J. marc, Taigbenu, A. E., & Boroto, J. R. (2007). Domestic rainwater harvesting to improve water supply in rural South Africa. *Physics and Chemistry of the Earth*, *32*(15–18), 1050–1057. https://doi.org/10.1016/j.pce.2007.07.007

Mwenge Kahinda, J., Taigbenu, A. E., & Boroto, R. J. (2010). Domestic rainwater harvesting as an adaptation measure to climate change in South Africa. *Physics and Chemistry of the Earth*, *35*(13–14), 742–751. https://doi.org/10.1016/j.pce.2010.07.004

Mwenge Kahinda, J., Taigbenu, A. E., Sejamoholo, B. B. P., Lillie, E. S. B., & Boroto, R. J. (2009). A GIS-based decision support system for rainwater harvesting (RHADESS). *Physics and Chemistry of the Earth*, *34*(13–16), 767–775.

https://doi.org/10.1016/j.pce.2009.06.011

Myers, N., R.A. Mittermeier, C.G. Mittermeier, G.A.B. da Fonseca & J. Kent (2000). Biodiversity hotspots for conservation priorities. Nature 403, 853–858

Nthuni, S., Lübker, T., & Schaab, G. (2014). Modelling the potential of rainwater harvesting in western Kenya using remote sensing and GIS techniques. *South African Journal of Geomatics*, *3*(3), 285–301. Retrieved from http://www.ajol.info/index.php/sajg/article/view/110380

Ojwang, R. O., Dietrich, J., Anebagilu, P. K., Beyer, M., & Rottensteiner, F. (2017). Rooftop rainwater harvesting for Mombasa: Scenario development with image classification and water resources simulation. *Water (Switzerland)*, *9*(5). https://doi.org/10.3390/w9050359

Priscoli, J. D. (2004). What is Public Participation in Water Resources Management and Why is it Important? *Water International*, *29*(2), 221–227. https://doi.org/10.1080/02508060408691771

Recha, C. W., Mukopi, M. N., & Otieno, J. O. (2015). Socio-Economic Determinants of Adoption of Rainwater Harvesting and Conservation Techniques in Semi-Arid Tharaka Sub-County, Kenya. *Land Degradation and Development*. https://doi.org/10.1002/ldr.2326

Shadeed, S., & Lange, J. (2010). Rainwater harvesting to alleviate water scarcity in dry conditions: A case study in Faria Catchment, Palestine. *Water Science and Engineering*, *3*(2), 132–143. https://doi.org/DOI: 10.3882/j.issn.1674-2370.2010.02.002

Shi, W., & Cheung, C. (2006). Performance Evaluation of Line Simplification Algorithms for Vector Generalization. *The Cartographic Journal*, *43*(1), 27–44. https://doi.org/10.1179/000870406X93490

Shinde, S. D., & Gaikwad, V. P. (2016). Application of Gis for Mapping Rainwater Harvesting Potential : a Case Study of Nidhal Village in Satara District , Maharashtra , India, *4*(5), 141–148.

Siegert, K., Chapman, C., & Finkel, M. (1991). WATER HARVESTING. *A Manual for the Design and Construction of Water Harvesting Schemes for Plant Production*. https://doi.org/10.1017/S1366728998000133

Soini, E. (2005). Livelihoods, capital, strategies and outcomes in the Taita Hills of Kenya.

Unpublished project report. 34 p

Thomas, T. H., & Martinson, D. B. (2007). *Roofwater harvesting*.

Traboulsi, H., & Traboulsi, M. (2017). Rooftop level rainwater harvesting system. *Applied Water Science*, *7*(2), 769–775. https://doi.org/10.1007/s13201-015-0289-8

Visvalingam, M. and Whyatt, J. D. (1993). 'Line generalization by repeated elimination of points',

The Cartographic Journal, 30, 46–51.

Visvalingam, M. and Williamson, P. J. (1995). 'Simplification and generalization of large scale data for roads: a comparison of two filtering algorithms', Cartography and Geographic Information Systems, 22(4), 264–275

Wang, O., Lodha, S. K., & Helmbold, D. P. (2007). A Bayesian approach to building footprint extraction from aerial LIDAR data. *Proceedings - Third International Symposium on 3D Data Processing, Visualization, and Transmission, 3DPVT 2006*, 192–199. https://doi.org/10.1109/3DPVT.2006.9

Weibel, R. (1997). 'Generalization of spatial data: principles and selected algorithms', in Algorithmic Foundations of Geographic Information Systems, ed. by van Kreveld, M., Nievergelt, J., Roos, T. and Widmayer, P., pp. 99–152, Springer-Verlag, Berlin

Zain M. Al-Houri, Oday K. Abu-Hadba, & Khaled A. Hamdan. (2014). The potential of roof top rainwater harvesting as a water resource in Jordan : Featuring two application case studies. *Environmental, Chemical, Ecological and Geophysical Engineering*, *8*(2), 147–153.

Zende, A. (2015). Rooftop rainwater harvesting system - a model based approach., (March), 1–4.

Zeng, C., Wang, J., & Lehrbass, B. (2013). An evaluation system for building footprint extraction from remotely sensed data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *6*(3), 1640–1652. https://doi.org/10.1109/JSTARS.2013.2256882

Zhe Li, Fergal Boyle, A. R. D. (2010). Rainwater Harvesting and Greywater Treatment Systems for Domestic Application in Ireland. *Pollution Research*, *33*(3), 597–603. https://doi.org/10.1016/j.desal.2010.05.035

Ziadat, F. M., Mazahreh, S. S., & Oweis, T. Y. (2006). A GIS-based Approach for Assessing

Water Harvesting Suitability in a Badia Benchmark Watershed in Jordan, *2006*(Isco), 1–4.

Zeng, C., Wang, J., & Lehrbass, B. (2013). An evaluation system for building footprint extraction from remotely sensed data. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 6(3), 1640–1652. https://doi.org/10.1109/JSTARS.2013.2256882

APPENDICES

Lastools in Command Line Windows:

```
set PATH=%PATH%;C:\LAStools\bin

lasoverage -i E:\Blayz_lidar\2015\LiDAR\Data\*.las ^
             -step 2  -remove_overage ^
             -odir E:\LIDAR_FINAL\2015\overage_removed -o taita.las

lastile -i E:\LIDAR_FINAL\2015\overage_removed\taita*.las ^
        -tile_size 1000 -buffer 30 -flag_as_withheld ^
          -files_are_flightlines ^
        -odir E:\LIDAR_FINAL\2015\lastiles_with_buffer -olas

lasnoise -i E:\LIDAR_FINAL\2013\lastiles_with_buffer\taita*.las^
          -step_xy 4 -step_z 2 -isolated 5 ^
         -odir E:\LIDAR_FINAL\2013\tiles_denoised -olas ^
         -cores 4

lasground -i E:\LIDAR_FINAL\2013\tiles_denoised\taita*.las ^
          -ignore_class 7 ^
          -town -fine ^
            -compute_height ^
          -odir E:\LIDAR_FINAL\2015\ground -olas ^
          -cores 4

lasheight -i E:\LIDAR_FINAL\2015\ground\taita*.las ^
          -drop_above 50 ^
            -drop_below 2 ^
          -odir E:\LIDAR_FINAL\2015\height -olas ^
          -cores 4

lasclassify -i E:\LIDAR_FINAL\2015\ground\taita*.las ^
            -ignore_class 7 ^
            -odir E:\LIDAR_FINAL\2015\class -olas ^
            -cores 4

lastile -i E:\LIDAR_FINAL\2015\class\taita*.las ^
        -remove_buffer ^
        -odir E:\LIDAR_FINAL\2015\lastiles_no_buffer -olas ^
        -cores 4

lasboundary -i E:\LIDAR_FINAL\2015\lastiles_no_buffer\taita*.las -merged ^
            -keep_class 6 ^
            -disjoint -concavity 1.4 ^
            -o E:\LIDAR_FINAL\2015\buildings\buildings_2015.shp
```

UTILITY FUNCTIONS

```
1. import rasterio
2. from rasterio.plot import show
3. from rasterio.plot import show_hist
4. from rasterio.mask import mask
5. from shapely.geometry import box
6. import geopandas as gpd
7. from fiona.crs import from_epsg
```

```
from osgeo import gdal
import utm
from gdalconst import GA_ReadOnly
import os
from shapely.geometry import Polygon
import numpy as np
from rasterio.features import shapes
import glob
import pandas as pd
import re


'''''
Author: Oyelowo Oyedayo
Purpose: For my thesis
Contact: www.github.com(Oyelowo)
'''

def get_density_spacing_info(input_dir):
    filepaths = os.path.join(input_dir, '*.txt')
    list_of_files = glob.glob(filepaths)
    lidar_info = pd.DataFrame(columns=['tilesNumber'])
    for i, file in enumerate(list_of_files, 1):
        with open(file, 'r') as f:
            text_lines = f.readlines()

    # get the point density and point spacing which are on line 39 and 40 respectively

        for line in text_lines:
          line=line.strip()
    # extract all and last return densities. Do same for spacing. There are two values for each
          if line.startswith('point density'):
            all_returns_density, last_returns_density = re.findall("\d+\.\d+", line)
          elif line.startswith('spacing'):
            all_returns_spacing, last_returns_spacing = re.findall("\d+\.\d+", line)
    # Insert the values into the dataframe
        lidar_info= lidar_info.append({
            'tilesNumber':int(i),
            'last_returns_density':float(last_returns_density),
            'all_returns_density': float(all_returns_density),
            'all_returns_spacing': float(all_returns_spacing),
            'last_returns_spacing': float(last_returns_spacing)
          },
          ignore_index=True)
    return lidar_info


def create_dir(dirName):
# Create target directory & all intermediate directories if don't exists
  if not os.path.exists(dirName):
    os.makedirs(dirName)
    print("Directory " , dirName ,  " Created ")
```

```
  else:
    print("Directory " , dirName ,  " already exists")
  return dirName


def bbox_to_utm(bbox, zone_number):
    lonlow, lathigh, lonhigh, latlow = bbox
    minx, maxy,*others = utm.from_latlon(lathigh, lonlow, zone_number)
    maxx, miny, *others = utm.from_latlon(latlow, lonhigh, zone_number)
    return [minx, maxy , maxx, miny]


def get_raster_extent(raster_file_path):
    data = gdal.Open(raster_file_path, GA_ReadOnly)
    geoTransform = data.GetGeoTransform()
    minx = geoTransform[0]
    maxy = geoTransform[3]
    maxx = minx + geoTransform[1] * data.RasterXSize
    miny = maxy + geoTransform[5] * data.RasterYSize
    print('[minx, maxy , maxx, miny] is',  [minx, maxy , maxx, miny] )
    data = None
    return [minx, maxy , maxx, miny]


def get_vector_extent(shapefile='', geometry_field='geometry', utm_zone=37, northern=Fa
lse):
    '''''
    extent: An array of the extent of the window in this order: [minx, maxy , maxx, min
y]
    '''
    minx, miny, maxx, maxy = shapefile[geometry_field].total_bounds
    latlow, lonlow = utm.to_latlon(minx, miny, utm_zone, northern=northern)
    lathigh, lonhigh = utm.to_latlon(maxx, maxy, utm_zone, northern=northern)
    return [lonlow, lathigh, lonhigh, latlow]


def clip_and_export_raster(raster_path, output_tif, extent):
    '''''
    extent: An array of the extent of the window in this order: [minx, maxy , maxx, min
y]
    '''
    raster_data = gdal.Open(raster_path)
    raster_data = gdal.Translate(output_tif, raster_data, projWin=extent)
    raster_data = None
           clipped = rasterio.open(output_tif)
           #show((clipped, 1), cmap='Blues')
           return clipped


      def getFeatures(gdf):
          """Function to parse features from GeoDataFrame in such a manner that raster
io wants them"""
```

```
            import json
            return [json.loads(gdf.to_json())['features'][0]['geometry']]
        
        
        
        
    def get_clipped_raster(raster_data, output_path, extent, bbox_epsg_code=4326):
            '''''
            extent: An array of the extent of the window in this order: [minx, maxy , ma
    xx, miny]
            '''
            minx, maxy , maxx, miny = extent
            bbox = box(minx, maxy , maxx, miny)

            geo = gpd.GeoDataFrame({'geometry': bbox}, index=[0], crs=from_epsg(bbox_eps
    g_code))

            try:
                geo = geo.to_crs(crs=raster_data.crs.data)
            except ValueError:
                print('The raster crs is not defined')
            coords = getFeatures(geo)

            clipped_img, clipped_img_transform = mask(raster=raster_data, shapes=coords,
     crop=True)
            clipped_img_meta = raster_data.meta.copy()
            clipped_img_meta.update({"driver": "GTiff",
                      "height": clipped_img.shape[1],
                        "width": clipped_img.shape[2],
                         "transform": clipped_img_transform,
                        "crs": raster_data.crs.data})

            with rasterio.open(output_path, "w", **clipped_img_meta) as output:
                output.write(clipped_img)

            clipped = rasterio.open(output_path)
            show((clipped, 1), cmap='terrain')
            return clipped
        
        
        
    def create_grid(gridHeight, gridWidth,bbox=None, is_utm=False, zone_number=None,
    shapefile=None,convex_hull=False, geometry_field='geometry'):
            '''''
            NOTE: you have to specify if your grid is in UTM or WGS84 longitude latitude

            bbox: should be provided
            '''
            if bbox is None and shapefile is None:
                raise ValueError('Provide either the bounding box or the shapefile you w
    ant to use for the extent of the grid')

            if bbox:
```

```
155.                 if not is_utm:
156.                     bbox=bbox_to_utm(bbox, zone_number)
157.                 minx, maxy , maxx, miny = bbox
158.             else:
159.                 minx,miny,maxx,maxy =  shapefile[geometry_field].total_bounds
160.                 print(shapefile[geometry_field].total_bounds)
161.
162.
163.             rows = int(np.ceil((maxy-miny) /  gridHeight))
164.             cols = int(np.ceil((maxx-minx) / gridWidth))
165.             XleftOrigin = minx
166.             XrightOrigin = minx + gridWidth
167.             YtopOrigin = maxy
168.             YbottomOrigin = maxy- gridHeight
169.
170.             polygons = []
171.             for i in range(cols):
172.                 Ytop = YtopOrigin
173.                 Ybottom =YbottomOrigin
174.                 for j in range(rows):
175.                     polygons.append(Polygon([(XleftOrigin, Ytop), (XrightOrigin, Ytop),
    (XrightOrigin, Ybottom), (XleftOrigin, Ybottom)]))
176.                     Ytop = Ytop - gridHeight
177.                     Ybottom = Ybottom - gridHeight
178.                 XleftOrigin = XleftOrigin + gridWidth
179.                 XrightOrigin = XrightOrigin + gridWidth
180.
181.             grid = gpd.GeoDataFrame({'geometry':polygons})
182.             if convex_hull:
183.               grid['geometry'] = grid.convex_hull
184.             return grid
185.
186.
187.
188.         def polygonize(raster_filepath, old_epsg_code=4326, new_epsg_code=32737):
189.             mask = None
190.             with rasterio.drivers():
191.                 with rasterio.open(raster_filepath) as original_raster:
192.                     image = original_raster.read(1) # first band
193.                     results = (
194.                     {'properties': {'grid_value': value}, 'geometry': geometry}
195.                     for index, (geometry, value) in enumerate(shapes(image, mask=mask, t
    ransform=original_raster.affine)))
196.
197.             geoms = list(results)
198.             gpd_polygonized_raster  = gpd.GeoDataFrame.from_features(geoms)
199.             gpd_polygonized_raster.crs = {'init' :'epsg:'+ str(old_epsg_code)}
200.             gpd_polygonized_raster = gpd_polygonized_raster.to_crs(epsg=new_epsg_code)
201.     return gpd_polygonized_raster
```

```
# ============================================================================
import geopandas as gpd
import matplotlib.pyplot as plt
from shapely.geometry import Polygon
import rasterio
from rasterio.plot import show
from rasterio.plot import show_hist
import seaborn as sns
import glob
import os
import calendar
import re
from mpl_toolkits.axes_grid1 import make_axes_locatable
import pysal as ps
from shapely import speedups
from pathlib import Path
speedups.available
speedups.enable()
```

```
# Access the utility functions I created earlier
```

```
import oyelowo_ras as ras

#my_path = os.path.abspath(os.path.dirname('__file__'))
my_dir = r'E:\LIDAR_FINAL\data'


ras.create_dir(my_dir)

def create_path(sub_dir='', my_dir=my_dir):
  return ras.create_dir(Path(my_dir + sub_dir))


aoi_dir = os.path.join(my_dir, 'AOI')
aoi_filepath =os.path.join(aoi_dir, 'fishnet_926_1sqm.shp')
bbox_raster_filepath = os.path.join(aoi_dir, 'clipped_mean_annual_rain.tif')
aoi_poly_filepath = os.path.join(aoi_dir, 'AOI_polygon.shp')
aoi_vertices_filepath= os.path.join(aoi_dir, 'aoi_vertices.shp')
buildings_filepath = os.path.join(my_dir,  'buildings', '2015', 'roof_polygons', 'buildings_2015_simplified.shp')

#rain_rasters_dir = ras.create_dir(os.path.join(my_dir,'precipitation'))
#output_clipped_raster_dir = ras.create_dir(os.path.join(my_dir, 'precipitation', 'clipped'))
#monthly_rain_shp_dir= ras.create_dir(os.path.join(my_dir, 'precipitation', 'clipped', 'to_vector'))
#centroid_filepath = ras.create_dir(os.path.join(my_dir,  'buildings', '2015', 'buildings_centroid', 'buildings_centroid.shp'))
#grid_filepath = ras.create_dir(os.path.join(my_dir,  'grid', 'grid.shp'))
#aoi_grid_clipped_shp_filepath = ras.create_dir(os.path.join(my_dir,  'grid', 'aoi_grid_clipped.shp'))


rain_rasters_dir = create_path('/precipitation')
output_clipped_raster_dir = create_path('/precipitation/clipped')
monthly_rain_shp_dir= create_path('/precipitation/clipped/to_vector')
centroid_filepath = create_path('/buildings/2015/buildings_centroid/buildings_centroid.shp')
```

```
grid_filepath = create_path('/grid/grid.shp')
aoi_grid_clipped_shp_filepath = create_path('/grid/aoi_grid_clipped.shp')




# ============================================================================
#
# ============================================================================
aoi_crs_epsg = {'init' :'epsg:32737'}
aoi_crs_epsg_code = 32737
rain_raster_data_epsg_code = 4326

#readthe shapefile for the area of interest
aoi_shapefile = gpd.read_file(aoi_filepath)


#bbox_aoi2 = ras.get_vector_extent(aoi_shapefile)'
#bbox_aoi = ras.get_vector_extent(aoi_shapefile)
#bbox_aoi = ras.get_raster_extent(bbox_raster_filepath)
bbox_aoi = [38.19986023835, -3.2418059025499986, 38.52486023705, -3.516805901449999]
aoi_polygon =  gpd.read_file(aoi_poly_filepath)
aoi_polygon.crs = aoi_crs_epsg
#bbox_aoi = ras.get_vector_extent(aoi_polygon)


# ============================================================================
# # CLIP ALL THE MONTHLY DATA AND ALSO SUM THEM
# ============================================================================
sum_rain = 0
rain_raster = glob.glob(os.path.join(rain_rasters_dir, '*.tif'))
for i, month_file_path in enumerate(rain_raster, 1):
    print(i)
    filename = os.path.basename(month_file_path)
#    Match the file name, excluding the extension name
    if filename[:-4] == 'annual_rainfall':
        month_name = 'ann'
    else:
        month_number = re.search(r'\d+', filename).group()
        month_name = calendar.month_name[int(month_number)]
    month_abbreviation = month_name[:3]+'_rain'
    output_tif = os.path.join(output_clipped_raster_dir, month_abbreviation + '.tif')
    print(output_tif)
    ras.clip_and_export_raster(month_file_path, output_tif, bbox_aoi)

    month_raster = rasterio.open(output_tif).read().astype(float)
    sum_rain += month_raster


       cc = sum_rain/12
       # ============================================================================

       # ['nearest' | 'bilinear' | 'bicubic' | 'spline16' |
       #            'spline36' | 'hanning' | 'hamming' | 'hermite' | 'kaiser' |
       #            'quadric' | 'catrom' | 'gaussian' | 'bessel' | 'mitchell' |
       #            'sinc' | 'lanczos' | 'none' ]
       # ============================================================================

       show(sum_rain,cmap='RdBu',interpolation="sinc", title="Mean Annual Rainfall")
```

```
109.        #show(month_raster,ax =ax, cmap='RdBu',interpolation="bessel", title="Mean Annua
    l Rainfall")
110.        # ================================================================================

111.        # CONVERT THE RASTER FILES INTO VECTOR
112.        # ================================================================================

113.        monthly_rain_clipped=glob.glob(os.path.join(output_clipped_raster_dir, '*.tif'))

114.        for i, month_file in enumerate(monthly_rain_clipped, 1):
115.            month_field_name = os.path.basename(month_file)[:3] + '_rain'
116.            output_shp = os.path.join(monthly_rain_shp_dir, month_field_name + '.shp')
117.            print(month_field_name)
118.        #    month_raster = rasterio.open(month_file)
119.            polygonized_raster = ras.polygonize(month_file, rain_raster_data_epsg_code,
    aoi_crs_epsg_code)
120.            polygonized_raster=polygonized_raster.rename(columns={'grid_value': month_fi
    eld_name})
121.            polygonized_raster.to_file(output_shp)
122.
123.        #    FOR TEST PURPOSE
124.            month_rain_data_test = gpd.read_file(output_shp)
125.            month_rain_data_test.plot(column=month_field_name, cmap="Blues", scheme="equ
    al_interval", k=9, alpha=0.9)
126.
127.
128.
129.        # ================================================================================

130.        # AOI POLYGON
131.        # ================================================================================

132.        # ================================================================================

133.        vertices = gpd.read_file(aoi_vertices_filepath)
134.        vertices.plot()
135.        aoi_vertices_list = [p.xy for p in vertices.geometry]
136.        aoi_polygon = Polygon([[points.x, points.y] for points in vertices.geometry])
137.        aoi_polygon_df = gpd.GeoDataFrame(data=[aoi_polygon],  columns=['geometry'])
138.        aoi_polygon_df.crs= aoi_crs_epsg
139.        aoi_polygon_df.plot()
140.        print(aoi_polygon)
141.        # ================================================================================

142.
143.
144.
145.
146.        # ================================================================================

147.        # WORKING WITH THE BUILDING SHAPEFILE
148.        # ================================================================================

149.        buildings_shp_unfiltered = gpd.read_file(buildings_filepath)
150.
151.        # calculate area and centroid of the buildings
152.        buildings_shp_unfiltered['area'] = buildings_shp_unfiltered['geometry'].area
153.        print(len(buildings_shp_unfiltered))
154.        # filter roof areas lower than 10sqm or higher than 2000sqm
155.        buildings_shp = buildings_shp_unfiltered.loc[(buildings_shp_unfiltered['area']>1
    0) & (buildings_shp_unfiltered['area']<2000)]
```

```
156.        print(len(buildings_shp))
157.        # get the centroid of every building
158.        buildings_shp['centroid']= buildings_shp['geometry'].centroid
159.
160.
161.        buildings_centroid = buildings_shp.copy()
162.        buildings_centroid['geometry'] = buildings_shp['centroid']
163.        buildings_centroid = buildings_centroid.reset_index(drop=True)
164.
165.        #set ID for the filtered buildings. Start from one
166.        buildings_centroid['ID'] =  buildings_centroid.index + 1
167.        del buildings_centroid['centroid']
168.
169.
170.
171.        buildings_centroid.to_file(centroid_filepath)
172.        #buildings_centroid.plot()
173.
174.        # ============================================================================

175.        #
176.        # ============================================================================

177.
178.
179.
180.
181.
182.
183.        # ============================================================================

184.        #
185.        # # CREATE A FISHNET/GRID OF 926.1m PIXEL
186.        # ============================================================================

187.        #generating grid by directly providing the bounding box
188.        grid = ras.create_grid(926.1, 926.1, shapefile=buildings_centroid)
189.        #generating grid based on shapefile extent
190.        #grid2 = ras.create_grid(926.1, 926.1, shapefile=aoi_shapefile)
191.
192.        #grid = ras.create_grid(gridHeight=926.1, gridWidth=926.1,shapefile=aoi_shapefil
    e)
193.        #grid.plot()
194.
195.
196.        grid.to_file(grid_filepath)
197.
198.        # ============================================================================

199.        # CLIP THE GRID INTO THE AOI
200.        grid.crs = aoi_crs_epsg
201.        aoi_grid_clipped = gpd.overlay(grid, aoi_polygon_df, how='intersection')
202.
203.        #reset the index of the joined data and use the index values + 1, as the ID of e
    ach grid
204.        aoi_grid_clipped['grid_ID'] = aoi_grid_clipped.reset_index(drop=True).index + 1

205.        aoi_grid_clipped.plot()
206.        aoi_grid_clipped.to_file(aoi_grid_clipped_shp_filepath)
207.
208.        #grid = gpd.read_file(r'E:\LIDAR_FINAL\data\grid\grid_clipped.shp'
```

```
209.        # ================================================================================
210.        #
211.        # ================================================================================
212.
213.
214.
215.
216.
217.        # ================================================================================
218.        # SPATIAL JOIN
219.        # ================================================================================
220.        grid.crs = buildings_centroid.crs = aoi_crs_epsg
221.        #join the grid with the buildings, to get the areas per grid
222.        buildings_grid = gpd.sjoin(aoi_grid_clipped,buildings_centroid, how="left", op='
    intersects')
223.
224.        #some grids might be without buildings and will be nan values. Replace those wit
    h 0
225.        buildings_grid = buildings_grid.fillna(0)
226.
227.        #delete the index_right column to avoid issues later
228.        del buildings_grid['index_right']
229.
230.        months_shp_filepaths = glob.glob(os.path.join(monthly_rain_shp_dir,'*.shp'))
231.
232.
233.
234.        # ================================================================================
235.        # AGGREGATE ROOF AREAS BASED ON GRID ID
236.        # ================================================================================
237.
238.
239.
240.        buildings_grouped = buildings_grid.groupby('grid_ID')
241.        buildings_aggr = gpd.GeoDataFrame()
242.        #buildings_aggr['geometry']=None
243.        grid_ID, geom ,area, buildings_count = [], [], [], []
244.        for key, (i, group ) in enumerate(buildings_grouped,1):
245.            group_geometry = group.iloc[0]['geometry']
246.            grid_ID.append(key)
247.            geom.append(group_geometry)
248.            area.append(group['area'].sum())
249.            buildings_count.append(len(group))
250.            print('Aggregating grid', key,  'Total Area=', group['area'].sum(),'with', l
    en(group) ,' buildings')
251.        buildings_aggr['grid_ID'] = grid_ID
252.        buildings_aggr['geometry'] = geom
253.        buildings_aggr['area_sum'] = area
254.        buildings_aggr['buildings_count'] = buildings_count
255.
256.
257.
258.        # ================================================================================
259.        # AGGREGATE RAINFALL DATA
```

```
260.        # ================================================================================

261.        #3CREATE FUNCTION TO HELP WITH AGGREGATING THE DATA
262.        #test['geometry'] = test.centroid
263.
264.        def aggregate_grid_rain(new_dataframe, old_dataframe, month_field_name):
265.            grouped_data = old_dataframe.groupby('grid_ID')
266.            #buildings_aggr['geometry']=None
267.            grid_ID_list, geometry_list, total_grid_rain_list , roof_area_list, building
    s_count_list =[], [], [], [], []
268.            for key, (i, group ) in enumerate(grouped_data,1):
269.                group_geometry = group.iloc[0]['geometry']
270.                grid_ID_list.append(key)
271.                geometry_list.append(group_geometry)
272.                total_grid_rain_list.append(round(group[month_field_name].mean(), 2))
273.                roof_area_list.append(group['area_sum'].sum())
274.                buildings_count_list.append(group.buildings_count.mean())
275.                print('Aggregating', key, month_field_name, group[month_field_name].mean
    ())
276.
277.            new_dataframe['area_sum'] = roof_area_list
278.            new_dataframe['geometry'] = geometry_list
279.            new_dataframe['grid_ID'] =grid_ID_list
280.            new_dataframe[month_field_name] = total_grid_rain_list
281.            new_dataframe['buildings_count'] = buildings_count_list
282.            return new_dataframe
283.
284.
285.        # ================================================================================

286.        # SPATIAL JOIN OF RAINFALL AND ROOF AREAS TO GRID DATA
287.        # ================================================================================

288.        import time
289.        start_time = time.time()
290.
291.
292.
293.        months_shp_filepaths = glob.glob(os.path.join(monthly_rain_shp_dir, '*.shp'))
294.
295.
296.        buildings_rain_aggr = gpd.GeoDataFrame()
297.        #del buildings_rain['area']
298.        for i, month_filepath in enumerate(months_shp_filepaths, 1):
299.            print(i)
300.            month_rain_data = gpd.read_file(month_filepath)
301.
302.            buildings_aggr.crs = month_rain_data.crs=  aoi_crs_epsg
303.
304.            joined_data = gpd.sjoin(buildings_aggr, month_rain_data, how='left', op='int
    ersects')
305.
306.        #    Get field name from file name and exclude the file format
307.            month_field_name = os.path.basename(month_filepath)[:-4]
308.            print(month_field_name)
309.
310.            buildings_rain_aggr = aggregate_grid_rain(buildings_rain_aggr, joined_data,
    month_field_name)
311.
312.        buildings_rain_aggr.plot(column='buildings_count' , legend=True)
313.        print("--- %s seconds ---" % (time.time() - start_time))
```

```
        # ================================================================================

        # PLOT THE ROOF AREA AND RAINFALL DATA
        # ================================================================================

        for column in buildings_rain_aggr.columns[3:]:
            buildings_rain_aggr.plot(column=column, cmap="Blues", scheme="quantiles", k=9, alpha=0.9)
            print(column)


        # ================================================================================

        # #From the fieldwork, the average water consumption per day is 136litres.
        # #The lowland and highland has very little difference, 134 to 138
        # ================================================================================

        daily_Water_use_per_home = 136
        monthly_wateruse_per_home = daily_Water_use_per_home * 30
        buildings_rain_aggr['monthly_water_use'] = monthly_wateruse_per_home * buildings_rain_aggr['buildings_count']
        buildings_rain_aggr['yearly_water_use'] = daily_Water_use_per_home * 365 * buildings_rain_aggr['buildings_count']

        buildings_rain_aggr.plot('area_sum', linewidth=0.03, cmap="YlOrBr", scheme="quantiles", k=19, alpha=0.9)
        buildings_rain_aggr.plot('buildings_count', linewidth=0.03, cmap="YlOrBr", scheme="quantiles", k=19, alpha=0.9)


        # ================================================================================

        #   CALCULATE MONTHLY RAINWATER HARVESTING POTENTIALS AND POTENTIAL MINUS
        #   WATER USE PER GRID TO SEE IF IT IS ENOUGH
        # ================================================================================

        #buildings_rain_aggr = buildings_rain_aggr.fillna(0)
        roof_coefficient = 0.7
        for column in buildings_rain_aggr.columns:
            if column in ['ann_rain','Apr_rain','Aug_rain','Dec_rain','Feb_rain','Jan_rain','Jul_rain','Jun_rain','Mar_rain','May_rain','Nov_rain','Oct_rain','Sep_rain']:
                print(column)
                roof_area = buildings_rain_aggr['area_sum']
                rainfall = buildings_rain_aggr[column]
                roof_harvesting_potential = (roof_area * rainfall * roof_coefficient)
                rain_pot = round(roof_harvesting_potential, 2)
                buildings_rain_aggr[column + 'POT'] = rain_pot
                if column == 'ann_rain':
                  buildings_rain_aggr['ann_pot_vs_use'] = rain_pot - buildings_rain_aggr.yearly_water_use
                  buildings_rain_aggr['ann_pot_vs_use_class'] = buildings_rain_aggr.ann_pot_vs_use.apply(lambda x: 'positive' if x>=0 else 'negative')
                  continue
            #    1 m2 * 1 mm = 1litre. roof area is m2 and rain is in mm.
                buildings_rain_aggr[column[:3] + '_pot_vs_use'] = rain_pot - buildings_rain_aggr.monthly_water_use
                buildings_rain_aggr[column[:3] + '_pot_vs_use_class'] = buildings_rain_aggr[column[:3] + '_pot_vs_use'].apply(lambda x: 'positive' if x>=0 else 'negative')


```

```
359.
360.       # ================================================================================
361.       # PLOT ROOF HARVESTING POTENTIAL FOR ALL MONTHS
362.       # ================================================================================
363.       for column in buildings_rain_aggr.columns:
364.           if column.endswith('rainPOT'):
365.               ax=grid.plot()
366.               buildings_rain_aggr.plot(ax=ax,column=column, cmap="RdBu", scheme="quant
    iles", k=10, alpha=0.9, edgecolor='1')
367.
368.               print(column)
369.       #edgecolor='0.8
370.       buildings_rain_aggr.describe()
371.
372.
373.       # ================================================================================
374.       # PLOTTING MAPS
375.       # ================================================================================
376.
377.
378.       def organise_colorbar(cbar, vmin, vmax, number_of_ticks=6, cbar_texts_padding=2.
    5, labelpad=17):
379.         #    organise the labels of the colorbar
380.           tick_padding=1.4
381.           interval = int((vmax-vmin)/(number_of_ticks-1))
382.           scale_divider = ((vmax-vmin)/interval)
383.           labels=[]
384.           for i in range(vmin, vmax+1, interval):
385.             labels.append(str(i))
386.       #    add greater than sign to the upper end of the scale
387.           labels[-1] = '>' + labels[-1]
388.           cbar.ax.set_yticklabels(labels)
389.           ticks = ['-'] * len(labels)
390.           for i, (tick, lab) in enumerate(zip(ticks , labels),0):
391.             cbar.ax.text(tick_padding, (i) / scale_divider, tick, ha='center', va='cen
    ter', weight='bold')
392.             if i == len(labels) -1:
393.               cbar_texts_padding += 0.75
394.             cbar.ax.text(cbar_texts_padding, (i) / scale_divider, lab, ha='center', va
    ='center')
395.       #    cbar.ax.get_yaxis().labelpad = labelpad
396.
397.
398.       def colorbar(ax, vmin, vmax, truncate_cbar_texts=True, cbar_title=None, number_o
    f_ticks=6, cbar_label_pad=2, labelpad = 17):
399.         # add colorbar
400.           fig = ax.get_figure()
401.           sm = plt.cm.ScalarMappable(cmap=rain_potential_cmap, norm=plt.Normalize(vmin
    =vmin, vmax=vmax))
402.           divider = make_axes_locatable(ax)
403.           cax = divider.append_axes("right", size="4%", pad=0.05)
404.           # fake up the array of the scalar mappable....
405.           sm._A = []
406.           cbar=fig.colorbar(sm, cax = cax, fraction=0.046)
407.           cbar.set_label(cbar_title, rotation=270)
408.           cbar.ax.get_yaxis().labelpad = labelpad
409.           if truncate_cbar_texts:
```

```
            cbar.ax.get_yaxis().set_ticks([])
            organise_colorbar(cbar, vmin, vmax)




      def find_month(column_name):
        month_list = ['January', 'February', 'March', 'April', 'May', 'June', 'July',
                      'August', 'September', 'October', 'November', 'December']
        month_abbreviation= column_name[:3]
        print(month_abbreviation)
      #  month = filter(lambda x: x.startswith(month_abbreviation), month_list)
        month = [month for month in month_list if month.startswith(month_abbreviation)
]
        return ' '.join(month)



      def userDefinedClassifer(class_lower_limit, class_upper_limit, class_step):
        breaks = [x for x in range(class_lower_limit, class_upper_limit, class_step)]

        classifier = ps.User_Defined.make(bins=breaks)
        return classifier


      def plot_map(dataFrame, column_list, scale_cmaps, vmin, vmax,truncate_cbar_texts
, l_limit, h_limit, step, output_fp, main_title, cbar_title, labelpad):
        fig, axes = plt.subplots(4, 3, figsize=(12,12), sharex=True, sharey=True)
      #  plt.suptitle('RAINWATER HARVESTING POTENTIAL IN TAITA')
      #  vmin, vmax = dataFrame[column_list].min().min(), dataFrame[column_list].max()
.max()
        classified_df = dataFrame.copy()
        classified_df[column_list] = classified_df[column_list].apply(userDefinedClass
ifer(l_limit, h_limit, step))
        plt.suptitle(main_title, fontsize=18)
      #  plt.tight_layout()
        for i, (ax, column) in enumerate(zip(axes.flatten(), column_list), 1):
          #Join the classes back to the main data.
          month = find_month(column)
      #    print(month)
          if not scale_cmaps:
            vmin, vmax = dataFrame[column].min(), dataFrame[column].max()
            map_plot=dataFrame.plot(ax=ax, column=column,linewidth=0.02,scheme="equal_
interval", k=9, cmap=rain_potential_cmap,  alpha=0.9)
          map_plot=classified_df.plot(ax=ax, column=column,linewidth=0.02, cmap=rain_p
otential_cmap,  alpha=0.9)
          print(column)
          ax.grid(b=True, which='minor', color='#D3D3D3', linestyle='-')
          ax.set_aspect('equal')

          # Rotate the x-axis labels so they don't overlap
          plt.setp(ax.xaxis.get_majorticklabels(), rotation=20)
          map_plot.set_facecolor("#eeeeee")
          minx,miny,maxx,maxy =  dataFrame.total_bounds

          # these are matplotlib.patch.Patch properties
          props = dict(boxstyle='round', facecolor='#eaeaea', alpha=0)
          map_plot.text(x=minx+1000,y=maxy-
5000, s=u'N \n\u25B2 ', ha='center', fontsize=17, weight='bold', family='Courier new',
rotation = 0)
```

```
461.            map_plot.text(x=426000,y=maxy+2000, s=month,  ha='center', fontsize=20, weig
    ht='bold', family='Courier new', bbox=props)
462.            plt.setp(ax.xaxis.get_majorticklabels(), rotation=20)
463.            colorbar(map_plot, vmin, vmax, truncate_cbar_texts, cbar_title, labelpad=lab
    elpad)
464.            plt.subplots_adjust(top=0.92)
465.            plt.savefig(output_fp, bbox_inches='tight',dpi=300, pad_inches=0.1)
466.
467.
468.
469.
470.
471.
472.        month_list = ['January', 'February', 'March', 'April', 'May', 'June', 'July',
473.                      'August', 'September', 'October', 'November', 'December']
474.        rain_pot_list = list(map(lambda x: x[:3] + '_rainPOT', month_list))
475.        rain_pot_vs_use_list = list(map(lambda x: x[:3] + '_pot_vs_use', month_list))
476.        rain_pot_vs_use_class_list = list(map(lambda x: x[:3] + '_pot_vs_use_class', mon
    th_list))
477.        rain_list =list(map(lambda x: x[:3] + '_rain', month_list))
478.
479.        #Plot for monthly rainfall distribution
480.        rain_potential_cmap = 'Blues'
481.        monthly_main_title = "Monthly Distribution of Rainfall in Taita Region"
482.        monthly_rain_cbar_title = "mm"
483.        monthly_rain_output_fp = r'E:\LIDAR_FINAL\data\plots\jan_dec_rain_distribution_f
    inal4.jpg'
484.
485.        class_upper_limit = int(buildings_rain_aggr[rain_list].max().max())
486.        class_lower_limit = int(buildings_rain_aggr[rain_list].min().min())
487.
488.        plot_map(buildings_rain_aggr, rain_list, False, None , None, False, 0, 200, 1,
489.                 monthly_rain_output_fp, main_title= monthly_main_title,  cbar_title= mo
    nthly_rain_cbar_title, labelpad=15)
490.
491.
492.
493.        #Plot for monthly rainfall potential distribution
494.        monthly_main_title = "Spatio-
    temporal Distribution of Roof RainWater Harvesting Potential in Taita"
495.        monthly_rain_cbar_title = "100, 000 litres"
496.        monthly_rain_output_fp = r'E:\LIDAR_FINAL\data\plots\jan_dec_rain_Potential_dist
    ribution_final_1.jpg'
497.
498.
499.        rain_potential_cmap = 'RdYlBu'
500.        plot_map(buildings_rain_aggr, rain_pot_list, True, 0 , 5, True, 0, 500000, 1000,

501.                 monthly_rain_output_fp, main_title= monthly_main_title,  cbar_title= mo
    nthly_rain_cbar_title, labelpad=30)
502.        buildings_rain_aggr[rain_pot_list].max()
503.
504.
505.
506.
507.
508.
509.        buildings_rain_aggr.Nov_rain
510.        rain_potential_cmap = 'RdBu'
511.        plot_map(buildings_rain_aggr, rain_list[:6],0 , 193,True, 6, 193, 1, r'E:\LIDAR_
    FINAL\data\plots\jan_jun_rain.jpg')
```

```
512.        #plot_map(buildings_rain_aggr, rain_list[6:],0 , 193,False, 6, 193, 1,r'E:\LIDAR
    _FINAL\data\plots\jan_jun_rain.jpg')
513.        #plot_map(buildings_rain_aggr, rain_pot_list,0 , 5, True, 0, 500000, 1000, r'E:\
    LIDAR_FINAL\data\plots\jan_jun_RdYlBu__free_labelpadbet.jpg')
514.        #plot_map(buildings_rain_aggr, rain_pot_list[6:], 0, 5,True, 0, 500000, 1000, r'
    E:\LIDAR_FINAL\data\plots\jul_dec_RdYlBu__free_labelpadbeta.jpg')
515.        # ============================================================================

516.        #
517.        # ============================================================================

518.        plot_map(buildings_rain_aggr, 'ann_rain',0 , 5, True, 0, 500000, 1000, r'E:\LIDA
    R_FINAL\data\plots\annual_tight_potential.jpg')
519.        buildings_rain_aggr.plot(column='ann_rainPOT', cmap='RdBu', scheme="quantiles",
    k=10, alpha=0.9,edgecolor='0.6')
520.        plt.hist(buildings_rain_aggr['Sep_rainPOT'])
521.
522.
523.        # ============================================================================

524.        # Total Annual potential
525.        # ============================================================================

526.        def plot_annual(dataframe, column, map_title, legend_title,cmap, output_fp, cate
    gorical):
527.          minx, miny, maxx, maxy =  buildings_rain_aggr.total_bounds
528.          fig, ax = plt.subplots(figsize  = (9, 9))
529.          if categorical:
530.            map_plot = dataframe.plot(ax =ax,figsize=fig, column=column, categorical=Tru
    e,linewidth=0.02, cmap=cmap, alpha=0.9,legend = True)
531.          else:
532.            map_plot = dataframe.plot(ax =ax,figsize=fig, column=column,scheme='quantile
    s', k=9,linewidth=0.02, cmap=cmap, alpha=0.9,legend = True)
533.          ax.grid(b=True, which='minor', color='#D3D3D3', linestyle='-')
534.          ax.set_aspect('equal')
535.          map_plot.set_facecolor("#eaeaea")
536.          map_plot.text(x=minx,y=maxy-
    6000, s=u'N \n\u25B2 ', ha='center', fontsize=37, weight='bold', family='Courier new',
    rotation = 0)
537.          ax.get_legend().set_bbox_to_anchor((1, 0.61))
538.          #ax.get_legend().set_bbox_to_anchor((1.43, 0.8))
539.          ax.get_legend().set_title(legend_title)
540.          ax.get_figure()
541.          ax.set_aspect('equal')
542.          plt.xlim(minx-5000, maxx+20000)
543.          ax.set_title(map_title, fontsize=15)
544.          #plt.axis('equal')
545.          #plt.show()
546.          plt.savefig(output_fp,dpi=300,  bbox_inches='tight', pad_inches=0.1)
547.
548.
549.
550.
551.        cmap='Blues'
552.        map_title='Distribution of Total Annual Rainfall in Taita Region'
553.        legend_title='Rainfall(mm)'
554.        output_fp = r'E:\LIDAR_FINAL\data\plots\total_annual_rain_final_final_4'
555.        plot_annual(buildings_rain_aggr, 'ann_rain', map_title, legend_title,cmap, outpu
    t_fp)
556.
557.
```

```
        buildings_rain_aggr_ = buildings_rain_aggr.copy()
        buildings_rain_aggr_['ann_rainPOT'] = round((buildings_rain_aggr['ann_rainPOT']/
    1000),0).astype(int)

        cmap='RdYlBu'
        map_title='Distribution of Annual Roof Rainwater Harvesting in Taita Region'
        legend_title='RRWHP (thousand litres)'
        output_fp = r'E:\LIDAR_FINAL\data\plots\total_annual_rain_potential_final_final_
    9'
        plot_annual(buildings_rain_aggr_, 'ann_rainPOT', map_title, legend_title,cmap, o
    utput_fp, False)




        cmap='Oranges'
        map_title='Distribution of Areas of Roofs in Taita Region'
        legend_title='Area (sqm)'
        output_fp = r'E:\LIDAR_FINAL\data\plots\total_roof_areas_final_final_7.jpeg'
        plot_annual(buildings_rain_aggr_, 'area_sum', map_title, legend_title,cmap, outp
    ut_fp, False)



        cmap='RdYlBu'
        map_title='Comparison of RRWH Potential and Water Use'
        legend_title='RRWHP minus Water Use(litres)'
        output_fp = r'E:\LIDAR_FINAL\data\plots\annual_pot_vs_use_final_final.jpeg'
        plot_annual(buildings_rain_aggr_, 'ann_pot_vs_use', map_title, legend_title,cmap
    , output_fp, False)


        cmap='RdYlBu'
        map_title='Comparison of RRWH Potential and Water Use'
        legend_title='RRWHP minus Water Use'
        output_fp = r'E:\LIDAR_FINAL\data\plots\annual_pot_vs_use_final_final_class.jpeg
    '
        plot_annual(buildings_rain_aggr_, 'ann_pot_vs_use_class', map_title, legend_titl
    e,cmap, output_fp, True)

        # ============================================================================

        # RRWHP VS WATER USE
        # ============================================================================

        #Plot to see if potential meets needs, monthly
        monthly_rain_output_fp = r'E:\LIDAR_FINAL\data\plots\monthly_pot_vs_use_distribu
    tion_final.jpg'
        fig, axes = plt.subplots(4, 3, figsize=(10,12), sharex=True, sharey=True)
        plt.suptitle('Comparison of RRWH Potential and Water Use in Taita', fontsize=18)

        #  vmin, vmax = dataFrame[column_list].min().min(), dataFrame[column_list].max()
    .max()
        #  plt.tight_layout()
        for i, (ax, column) in enumerate(zip(axes.flatten(), rain_pot_vs_use_class_list)
    , 1):
          #Join the classes back to the main data.
          month = find_month(column)
        #    print(month)
          legend=False
          if i == 3:
```

```
606.           legend=True
607.         map_plot=buildings_rain_aggr.plot(ax=ax, column=column,linewidth=0.02,legend=l
    egend, cmap='RdYlBu',  alpha=0.9)
608.         print(column)
609.         ax.grid(b=True, which='minor', color='#D3D3D3', linestyle='-')
610.         ax.set_aspect('equal')
611.
612.         # Rotate the x-axis labels so they don't overlap
613.         plt.setp(ax.xaxis.get_majorticklabels(), rotation=20)
614.         map_plot.set_facecolor("#eeeeee")
615.         minx,miny,maxx,maxy =  buildings_rain_aggr.total_bounds
616.         maxx += 2500
617.         if i==3:
618.           map_plot.get_legend().set_bbox_to_anchor((1.8, 0.97))
619.           #ax.get_legend().set_bbox_to_anchor((1.43, 0.8))
620.           map_plot.get_legend().set_title('RRWHP vs Water Use')
621.         # these are matplotlib.patch.Patch properties
622.         props = dict(boxstyle='round', facecolor='#eaeaea', alpha=0)
623.         map_plot.text(x=minx+1000,y=maxy-
    5000, s=u'N \n\u25B2 ', ha='center', fontsize=17, weight='bold', family='Courier new',
    rotation = 0)
624.         map_plot.text(x=426000,y=maxy+2000, s=month,  ha='center', fontsize=20, weight
    ='bold', family='Courier new', bbox=props)
625.         plt.setp(ax.xaxis.get_majorticklabels(), rotation=20)
626.       #  plt.tight_layout()
627.         plt.subplots_adjust(top=0.92)
628.         plt.savefig(monthly_rain_output_fp, bbox_inches='tight',dpi=300, pad_inches=0.
    1)
629.       # ================================================================================

630.       #
631.       # ================================================================================

632.
633.
634.
635.       # ================================================================================

636.       # TOTAL ANNUAL POTENTIAL
637.       # ================================================================================

638.
639.       from matplotlib.pyplot import figure
640.       figure(num=None, figsize=(8, 5), dpi=80, facecolor='#eaeaea', edgecolor='k')
641.       first_letter = [first[:3] for first in rain_pot_list]
642.       plt.bar(first_letter, buildings_rain_aggr[ rain_pot_list].sum(), color='lightblu
    e')
643.       plt.plot(first_letter, buildings_rain_aggr[ rain_pot_list].sum(), 'p-')
644.       plt.ylim(0, buildings_rain_aggr[ rain_pot_list].sum().max() + 100000000)
645.       buildings_rain_aggr[rain_list].mean()
646.       plt.title('Total Monthly Roof Rainwater Harveting Potential, Taita')
647.       plt.xlabel('Months')
648.       plt.ylabel('RRHP\n(100 million litres)')
649.       plt.rcParams['axes.facecolor'] = '#ffffff'
650.       plt.savefig(r'E:\LIDAR_FINAL\data\plots\bar_line_RRWP_months_series2.jpeg', dpi=
    300, bbox_inches='tight', pad_inches=0.1)
651.
652.
653.
654.       # ================================================================================
```

```
655.        # PERCENTAGE OF BUILDINGS THAT RRWH CAN FULFILL THEIR NEEDS
656.        # ============================================================================

657.        #buildings_rain_aggr.to_file(r'E:\LIDAR_FINAL\data\aggregated\buildings_rain_aggr.shp')
658.
659.        percent_buildings_with_positive_RRWHP_list = []
660.        buildings_total_count = buildings_rain_aggr['buildings_count'].sum()
661.        for month_use_class in rain_pot_vs_use_class_list:
662.          gr = buildings_rain_aggr.groupby(month_use_class)
663.          buildings_with_net_positive_pot = (gr['buildings_count'].sum()['positive'] * 100) / buildings_total_count
664.          buildings_with_net_negative_pot = gr['buildings_count'].sum()['negative']
665.          percent_buildings_with_positive_RRWHP_list.append(buildings_with_net_positive_pot)
666.          print(percent_buildings_with_positive_RRWHP_list[-1])
667.
668.
669.
670.
671.        from matplotlib.pyplot import figure
672.        figure(num=None, figsize=(8, 5), dpi=80, facecolor='#eaeaea', edgecolor='k')
673.        first_letter = [first[:3] for first in rain_pot_list]
674.        plt.bar(first_letter, percent_buildings_with_positive_RRWHP_list, color='lightblue')
675.        plt.plot(first_letter,percent_buildings_with_positive_RRWHP_list , 'p-')
676.        #plt.ylim(0, buildings_rain_aggr[ rain_pot_list].sum().max() + 100000000)
677.        buildings_rain_aggr[rain_list].mean()
678.        plt.title('Percentage of Buildings in Taita that RRWH alone can Meet their Water Use')
679.        plt.xlabel('Months')
680.        plt.ylabel('Percentage of Buildings (%)')
681.        plt.rcParams['axes.facecolor'] = '#ffffff'
682.        plt.savefig(r'E:\LIDAR_FINAL\data\plots\bar_line_pot_vs_use_months1', dpi=300, bbox_inches='tight', pad_inches=0.1)
683.
684.
685.
686.
687.        buildings_rain_aggr.columns
688.        annual_use_class = 'ann_pot_vs_use_class'
689.        annual_group = buildings_rain_aggr.groupby('ann_pot_vs_use_class')
690.        annual_buildings_with_net_positive_pot = (annual_group['buildings_count'].sum()['positive'] * 100) / buildings_total_count
```

EXTRACT INFO FROM LIDAR

```
1.  import pandas as pd
2.  import matplotlib.pyplot as plt
3.  import glob
4.  import os
```

```
import re


def get_density_spacing_info(input_dir):
    filepaths = os.path.join(input_dir, '*.txt')
    list_of_files = glob.glob(filepaths)
    lidar_info = pd.DataFrame(columns=['tilesNumber'])
    for i, file in enumerate(list_of_files, 1):
        with open(file, 'r') as f:
            text_lines = f.readlines()

    # get the point density and point spacing which are on line 39 and 40 respectively

        for line in text_lines:
          line=line.strip()
    # extract all and last return densities. Do same for spacing. There are two values for each
          if line.startswith('point density'):
            all_returns_density, last_returns_density = re.findall("\d+\.\d+", line)
          elif line.startswith('spacing'):
            all_returns_spacing, last_returns_spacing = re.findall("\d+\.\d+", line)
    # Insert the values into the dataframe
        lidar_info= lidar_info.append({
            'tilesNumber':int(i),
            'last_returns_density':float(last_returns_density),
            'all_returns_density': float(all_returns_density),
            'all_returns_spacing': float(all_returns_spacing),
            'last_returns_spacing': float(last_returns_spacing)
          },
          ignore_index=True)
    return lidar_info


import clip_raster as ras
outputdir_2015 = r'E:\LIDAR_FINAL\data\lidar_tiles_info_output\lidar_info_2015.txt'
inputdir_2015 = r'E:\LIDAR_FINAL\data\lidar_tiles_info\2015'

outputdir_2013 = r'E:\LIDAR_FINAL\data\lidar_tiles_info_output\lidar_info_2013.txt'
inputdir_2013 = r'E:\LIDAR_FINAL\data\lidar_tiles_info\2013'

lidar2015_info = ras.get_density_spacing_info(inputdir_2015)
lidar2015_info.to_csv(outputdir_2015)

lidar2013_info = ras.get_density_spacing_info(inputdir_2013)
lidar2013_info.to_csv(outputdir_2013)
```

VISUALISE PULSE DENSITY AND SPACING

```
# Import library and dataset
import seaborn as sns
import matplotlib.pyplot as plt
import clip_raster as ras
```

```
from itertools import cycle


outputdir_2015 = r'E:\LIDAR_FINAL\data\lidar_tiles_info_output\lidar_info_2015.txt'
inputdir_2015 = r'E:\LIDAR_FINAL\data\lidar_tiles_info\2015'

outputdir_2013 = r'E:\LIDAR_FINAL\data\lidar_tiles_info_output\lidar_info_2013.txt'
inputdir_2013 = r'E:\LIDAR_FINAL\data\lidar_tiles_info\2013'

lidar2015_info = ras.get_density_spacing_info(inputdir_2015)
lidar2015_info.to_csv(outputdir_2015)

lidar2013_info = ras.get_density_spacing_info(inputdir_2013)
lidar2013_info.to_csv(outputdir_2013)

sns.boxplot(lidar2013_info['all_returns_density'])
sns.boxplot(lidar2015_info['all_returns_density'])
sns.boxplot(lidar2013_info['all_returns_spacing'])
sns.boxplot(lidar2015_info['all_returns_spacing'])



lidar2015_info.describe().to_csv(r'E:\LIDAR_FINAL\data\plots\2015_lidar_info_stat.txt')

lidar2013_info.describe().to_csv(r'E:\LIDAR_FINAL\data\plots\2013_lidar_info_stat.txt')


def plot_density_spacing(data, year, outut_fp):
  bg_color="#efefef"
  sns.set(rc={'axes.facecolor':bg_color, 'figure.facecolor':bg_color})
  fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12,8))

  ax11=axes[0, 0]
  ax12=axes[0, 1]
  ax21=axes[1, 0]
  ax22=axes[1, 1]

  plt.suptitle('Distribution of point density and spacing of the {year} LIDAR Data'.for
mat(year=year))
  # # Make default histogram of sepal length
  sns.distplot( data['all_returns_density'] , ax=ax11,  bins=20 ,rug=True)
  sns.distplot( data['last_returns_density'] , ax=ax12, bins=20, rug=True)
  sns.distplot( data['all_returns_spacing'] , ax=ax21, bins=20, rug=True)
  sns.distplot( data['last_returns_spacing'] , ax=ax22, bins=20, rug=True)

  ax11.set(xlabel='all returns point density(points/sqm)', ylabel='Fequency of point de
nsity')
  ax12.set(xlabel='last returns point density(points/sqm)', ylabel='')
  ax21.set(xlabel='all returns point spacing(m)', ylabel='Frequency of point spacing')

  ax22.set(xlabel='last returns point spacing(m),', ylabel='')
  plt.tight_layout()
  plt.subplots_adjust(top=0.95)
```

```
  plt.savefig(outut_fp, dpi=300,  bbox_inches='tight', pad_inches=0.1)


plot_density_spacing(lidar2013_info, '2013', 'E:/LIDAR_FINAL/data/plots/hist_lidar_density_spacing_2013')
plot_density_spacing(lidar2015_info, '2015', 'E:/LIDAR_FINAL/data/plots/hist_lidar_density_spacing_2015')


def plot_density_spacing_2(dataset, year, outut_fp):
  bg_color="#efefef"
  sns.set(rc={'axes.facecolor':bg_color, 'figure.facecolor':bg_color})
  fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(12,8))

  ax11=axes[0, 0]
  ax12=axes[0, 1]
  ax21=axes[1, 0]
  ax22=axes[1, 1]

  plt.suptitle('Comparison of the Distribution of the Point Density and Spacing of the {year} LIDAR Data'.format(year=year), fontsize=13)
  # # Make default histogram of sepal length
  for data, year in zip(dataset, cycle(['2013', '2015'])):
    sns.kdeplot( data['all_returns_density'] , ax=ax11, label='All Returns, ' + year, shade=True)
    sns.kdeplot( data['last_returns_density'] , ax=ax12, label='Last Returns, ' + year, shade=True)
    sns.kdeplot( data['all_returns_spacing'] , ax=ax21, label='All Returns, ' + year, shade=True)
    sns.kdeplot( data['last_returns_spacing'] , ax=ax22, label='Last Returns, ' + year, shade=True)

  ax11.set(xlabel='all returns point density(points/sqm)', ylabel='Fequency of point density')
  ax12.set(xlabel='last returns point density(points/sqm)', ylabel='')
  ax21.set(xlabel='all returns point spacing(m)', ylabel='Frequency of point spacing')

  ax22.set(xlabel='last returns point spacing(m),', ylabel='')
  plt.tight_layout()
  plt.subplots_adjust(top=0.95)
  plt.savefig(outut_fp, dpi=300, bbox_inches='tight', pad_inches=0.1)

plot_density_spacing_2([lidar2013_info, lidar2015_info],'2013 and 2015', 'E:/LIDAR_FINAL/data/plots/hist_lidar_density_spacing_combined')


```

AUTO EXTRACTED BUILDING FOOTPRINT POLYGONS VALIDATION

```
# ================================================================================
# VALIDATE ROOF AREAS
```

```
# ============================================================================

import geopandas as gpd
import matplotlib.pyplot as plt
from shapely.geometry import Polygon,Point

# ============================================================================
# IMPORT DATA
# ============================================================================
digitized_roof = gpd.read_file(r'E:\LIDAR_FINAL\data\building_digitised\digitizedb_bbox.shp')
roof_2013_all = gpd.read_file(r'E:\LIDAR_FINAL\data\buildings\2013\roof_polygons\buildings_2013_projected_regularized.shp')
roof_2015_all = gpd.read_file(r'E:\LIDAR_FINAL\data\buildings\2015\roof_polygons\buildings_2015_simplified.shp')
aoi = gpd.read_file(r'E:\LIDAR_FINAL\diigised_Samples_roofs\aoi_roof_samples.shp')



# ============================================================================
# GET THE BOUDING BOX OF THE DIGITISED ROOFS AND MAKE THE AOI OUT OF IT
# ============================================================================
xmin, ymin, xmax, ymax = digitized_roof.total_bounds
coords = [[xmin, ymin], [xmin, ymax],[xmax, ymax],[xmax, ymin]]
pp = Polygon([[point[0], point[1]] for point in coords])
aoi = gpd.GeoDataFrame(gpd.GeoSeries(pp), columns=['geometry'])
aoi.plot()


# ============================================================================
# SELECT ONLY ROOFS WITHIN THE AOI
# ============================================================================
aoi_boundary = aoi.loc[0].geometry
type(aoi_boundary)
roof_2013_all = roof_2013_all[roof_2013_all.geometry.within(aoi_boundary)]
roof_2013_all.plot()
roof_2015_all = roof_2015_all[roof_2015_all.geometry.within(aoi_boundary)]
roof_2015_all.plot()
len(roof_2013_all)
len(roof_2015_all)
len(digitized_roof)


# ============================================================================
# GET AREA AND CREATE ID FOR EACH POLYGON OF THE DIGITISED ROOFS
# ============================================================================
try:
  del digitized_roof['buildings']
  del digitized_roof['id']
  del digitized_roof['fid']
except:
  pass
digitized_roof['area'] = digitized_roof['geometry'].area
```

```
digitized_roof['digi_ID'] = digitized_roof.index + 1


# ============================================================================
# FILTER TOO SMALL OR TOO BIG POLYGONS
# ============================================================================
lower_limit, upper_limit = 10, 2000
roof_2013_all['area'] = roof_2013_all.geometry.area
roof_2013 = roof_2013_all.loc[(roof_2013_all['area']>lower_limit) & (roof_2013_all['area']<upper_limit)].reset_index(drop=True)

roof_2015_all['area'] = roof_2015_all.geometry.area
roof_2015 = roof_2015_all.loc[(roof_2015_all['area']>lower_limit) & (roof_2015_all['area']<upper_limit)].reset_index(drop=True)


# ============================================================================
# Check the data
# ============================================================================
roof_2013.plot(column='area', cmap="RdBu", scheme="quantiles", alpha=0.9)
roof_2013.isna().sum()
digitized_roof.isna().sum()

digitized_roof['geometry'][1]
roof_2013['geometry'][3]

roof_2013r = roof_2013.loc[6:12,:]
print(roof_2013.loc[12,'geometry'])



# ============================================================================
# INTERSECTING THE DIGITISED AND THE BUILDINGS EXTRACTED FROM LIDAR
# ============================================================================

digi_inter_roof13_df = gpd.sjoin(digitized_roof, roof_2013,how='inner',lsuffix='digi', rsuffix='lidar')
digi_inter_roof15_df = gpd.sjoin(digitized_roof, roof_2015,how='inner', lsuffix='digi', rsuffix='lidar')


# ============================================================================
# ACCURACY ANALYSIS: ERROR OF OMISSION
# ============================================================================
digi_roofs_count = len(digitized_roof)

#2013
digi_inter_roof13_df_grouped = digi_inter_roof13_df.groupby('digi_ID')
correct_roof2013_count = len(digi_inter_roof13_df_grouped)
omission_roof_2013 = ((digi_roofs_count - correct_roof2013_count) *100)/ digi_roofs_count

```

```
101.
102.       #2015
103.       digi_inter_roof15_df_grouped = digi_inter_roof15_df.groupby('digi_ID')
104.       correct_roof2015_count = len(digi_inter_roof15_df_grouped)
105.       omission_roof_2015 = ((digi_roofs_count - correct_roof2015_count) *100)/ digi_ro
   ofs_count
106.
107.       accuracy_statements = [
108.           'In 2013, {0} roofs were correctly extracted out of {1} roofs. \n'.format(co
   rrect_roof2013_count, digi_roofs_count),
109.           'In 2013, {0} roofs were omitted out of {1} roofs. \n'.format(digi_roofs_cou
   nt -correct_roof2013_count, digi_roofs_count),
110.            'Percentage of omitted buildings in 2013 is {:.2f}%. \n'.format(omission_ro
   of_2013),
111.            'Percentage accurately extracted in 2013 is {:.2f}%. \n\n'.format(100-
   omission_roof_2013),
112.
113.
114.             'In 2015, {0} roofs were rightly extracted out of {1} roofs'.format(correc
   t_roof2015_count, digi_roofs_count),
115.              'In 2015, {} roofs were omitted out of {} roofs. \n'.format(digi_roofs_co
   unt -correct_roof2015_count, digi_roofs_count),
116.             'Percentage of omitted buildings in 2015 is {:.2f}%. \n'.format(round(omis
   sion_roof_2015,2)),
117.             'Percentage accurately extracted in 2015 is {:.2f}%. \n\n'.format(100-
   omission_roof_2015)
118.             ]
119.
120.
121.       print(*accuracy_statements)
122.       # ================================================================================

123.       # ERROR OF COMMISSION
124.       # ================================================================================

125.       #2013
126.       #digi_inter_roof13_df_grouped = digi_inter_roof13_df.groupby('digi_ID')
127.       correct_roof2013_count = len(digi_inter_roof13_df)
128.       all_roofs_2013_count = len(roof_2013)
129.       falsely_extracted_2013 = (all_roofs_2013_count - correct_roof2013_count)
130.       commission_roof_2013 = (falsely_extracted_2013 *100)/ all_roofs_2013_count
          #2015
131.       correct_roof2015_count = len(digi_inter_roof15_df)
132.       all_roofs_2015_count = len(roof_2015)
133.       falsely_extracted_2015 = (all_roofs_2015_count - correct_roof2015_count)
134.       commission_roof_2015 = (falsely_extracted_2015 *100)/ all_roofs_2015_count
135.
136.
137.
138.       accuracy_statements2 = ['In 2013, {0} roofs were falsely extracted(false positiv
   e) out of {1} roofs extracted. \n'.format(falsely_extracted_2013 , all_roofs_2013_count
   ),
```

```
            'The percentage of falsely extracted is {:.2f}%. \n\n'.format(commission_roof_2013),


            'In 2015, {0} roofs were falsely extracted(false positive) out of {1} roofs extracted. \n'.format(falsely_extracted_2015 , all_roofs_2015_count),
            'The percentage of falsely extracted is {:.2f}%. \n'.format(commission_roof_2015)
            ]


      statement = '\n'
      for sentence in accuracy_statements:
        statement += sentence

      for sentence in accuracy_statements2:
        statement += sentence

      type(statement)
      print(statement)
      print(*accuracy_statements, *accuracy_statements2)

      #accuracy_metrics = str(*accuracy_statements, *accuracy_statements2)
      with open(r"E:\LIDAR_FINAL\data\plots\Output.txt", "w") as text_file:
          text_file.write(statement)




      # ================================================================================

      # ACCURACY: AREA, RMSE, MAE, AND SCATTERPLOT
      # ================================================================================

      roof_15_agg=gpd.GeoDataFrame()
      roof_15_agg['geometry'] = None
      for key, group in digi_inter_roof15_df_grouped:
        roof_15_agg.loc[key,'ID'] = key
        roof_15_agg.loc[key,'digi_area'] = group['area_digi'].unique()
        roof_15_agg.loc[key,'lidar_area'] =  group['area_lidar'].sum()
        roof_15_agg.loc[key,'one_to_N_rel'] = len(group['area_lidar'])
        print('Aggregating: ', key)


      from pandas.plotting import scatter_matrix
      import scipy
      import numpy as np
      roof_15_agg[23:25].plot()

      roof_15_agg.iloc[:,2:4].corr()
      scatter_matrix(roof_15_agg.iloc[:,2:4].corr())
      roof_15_agg_ = roof_15_agg.loc[(roof_15_agg['one_to_N_rel']==1) & (roof_15_agg.lidar_area<500)]
```

```
        x, y = roof_15_agg_.digi_area, roof_15_agg_.lidar_area
        slope, intercept, r_value, p_value, std_err = scipy.stats.linregress(x, y)
        r_value **2 * 100

        plt.scatter(x,y)
        plt.plot(np.unique(x), np.poly1d(np.polyfit(x, y, 1))(np.unique(x)))



        import seaborn as sns
        from scipy import stats
        def r2(x, y):
            return stats.pearsonr(x, y)[0] ** 2

        ax = sns.jointplot(x, y, kind="reg", stat_func=r2, logx=True, truncate=True, spa
    ce=0.1)
        plt.subplots_adjust(top=0.9)
        ax.fig.suptitle('ARoof Areas of Extracted vs Digitised', fontsize=20) # can also
     get the figure from plt.gcf()
        output_fp = r'E:\LIDAR_FINAL\data\plots\digi_vs_lidar2'
        plt.savefig(output_fp,  bbox_inches='tight',dpi=300, pad_inches=0.1)



        list(roof_15_agg.one_to_N_rel.astype(int)).count(1)

        from collections import Counter
        one_to_N_rel_frequency = Counter(list(roof_15_agg.one_to_N_rel.astype(int)))
        one_to_N_counts = one_to_N_rel_frequency.items()

        plt.hist(roof_15_agg.one_to_N_rel)
        plt.ylabel('Number of Buildings')
        plt.xlabel('One to N relationships')
        plt.title('One-to-One/One-to-
    Many Relationships Between Extracted Roofs \nand Digitized  Roofs')
        plt.ylim(0, 1500)
        for item in one_to_N_counts:
          plt.text(item[0]-0.05, item[1]+14, s=item[1])
        plt.savefig( r'E:\LIDAR_FINAL\data\plots\one_to_N_relationship1_2015_', bbox_inc
    hes='tight',dpi=300, pad_inches=0.1)


        ((x-y)/x) *100

        roof_15_agg_.mean()
        digi_inter_roof15_df_grouped['area_lidar'].agg(lambda x: print(x.mean()))


        import numpy as np

        def rmse(predictions, targets):
            return np.sqrt(((predictions - targets) ** 2).mean())

```

```
        rmse_val = rmse(x, y)
        print("rms error is: " + str(rmse_val))


        def smape(A, F):
            return 100/len(A) * np.sum(2 * np.abs(F - A) / (np.abs(A) + np.abs(F)))

        print(smape(x,y))
```

RAIN MODEL VALIDATION

```
# ================================================================================
# Validate rainfall data from CHELSA
# ================================================================================

import glob
import pandas as pd
import matplotlib.pyplot as plt
import os
import geopandas as gpd
from datetime import datetime
from shapely.geometry import Point
from scipy.stats import linregress
import seaborn as sns
from scipy import stats
from pathlib import Path
import clip_raster as ras

my_dir = r'E:\LIDAR_FINAL\data'


ras.create_dir(my_dir)

def create_path(sub_dir='', my_dir=my_dir):
  return ras.create_dir(Path(my_dir + sub_dir))

rain_dir = r"E:\LIDAR_FINAL\data\rainfall_data_field\rain"
rain_dir_all = r"E:\LIDAR_FINAL\data\rainfall_data_field\rain\Precipitation\*.XLSX"
buildings_rain_aggr = gpd.read_file(r'E:\LIDAR_FINAL\data\aggregated\buildings_rain_aggr.shp')
stations_filepath = r'E:\LIDAR_FINAL\data\rainfall_data_field\stations_locations\stations.csv'

rain_fp_list = glob.glob(rain_dir_all)

def aggregateDataByMonth(data):
  data.index = pd.to_datetime(data['Date'])
  data = data.groupby(pd.Grouper(freq="M"))
  df =pd.DataFrame()
  df['Date'] =None
  for key, group in data:
    df.loc[key, 'Date'] = key
```

```
    df.loc[key, 'rain_mm'] = group['Rain_(mm)'].sum()
#    print(key,'\n \n', group)
  return df.reset_index(drop=True)

all_data = pd.DataFrame(columns=['Date', 'rain_mm', 'station'])
for i, rain_data in enumerate(rain_fp_list[2:]):
  data = pd.read_excel(rain_data)
  header_index = data.loc[data['ID']=='Date'].index.values[0]
  data = pd.read_excel(rain_data, skiprows=header_index+1, parse_dates=["Date"])
  data = data[['Date', 'Rain_(mm)']]
  agg_data = aggregateDataByMonth(data)
  station_name = rain_data.split('\\')[-1].split('.')[0]
  agg_data['station'] = station_name.split('_Ar')[0]
  all_data = all_data.append(agg_data)
  print(agg_data.head(5))
  print(i)


# ================================================================================
# REMOVE THE ERRONEOUS DATA FROM KITUNKUYI
# ================================================================================
#The station errorneously recorded 0 from 7th August, 2014
kitunkyi_nodata_mask = pd.to_datetime(all_data['Date'].dt.date) <'2014-08-07'
all_data= all_data.loc[(all_data.station != 'Kitukunyi') | (kitunkyi_nodata_mask & (all
_data.station == 'Kitukunyi'))]

# ================================================================================


stations_names_list = all_data['station'].unique().tolist()

def rename_stations(station):
  if station == 'Taita_RS':
    station = "Taita Research Station"
  else:
    station += ' Weather Station'
  return station
#stations_list.remove('Mwatate_Ar112509')

def design_multi_plots(ax, station, ylim_min, ylim_max):
  ax.set_title(station, fontsize=18, weight='normal')
  ax.grid(color='grey', linestyle='--', linewidth=1, alpha=0.3)
  ax.set_ylim(ylim_min, ylim_max)
#  ax.set_facecolor('white')
  ax.tick_params(axis='both', which='major', labelsize=15)
  # Axis labels
  if i in [1, 3, 5]:
    ax.set_ylabel('Rainfall (mm)', fontsize=18, weight='normal')
  if i in [5, 6]:
    ax.set_xlabel('Date', fontsize=18, weight='normal')

plt.rcParams.update({'font.size': 20})
```

```
min_temp, max_temp = all_data.rain_mm.min()-20, all_data.rain_mm.max() + 20
fig, axes = plt.subplots(3, 2, figsize=(14,14), sharex=True)
for i, (ax, station) in enumerate(zip(axes.flatten(), stations_names_list), 1):
  sub_data = all_data.loc[all_data['station']==station]
  ax.plot(sub_data.Date, sub_data.rain_mm, lw = 1.5, c='blue')
  # Figure title
  fig.suptitle('Measured Rainfall in Taita Region')
  station = rename_stations(station)
        design_multi_plots(ax, station, min_temp, max_temp)
      plt.tight_layout()
      plt.subplots_adjust(top=0.92)
      plt.savefig(r'E:\LIDAR_FINAL\data\plots\stations_rain_timeseries.jpeg',  bbox_in
  ches='tight', pad_inches=0.1)



      monthly_agg_data = pd.DataFrame(columns=["station", "month", "rain_mm"])
      i=0
      for station in stations_names_list:
        sub_data = all_data.loc[all_data['station']==station]
        sub_data["month"] = sub_data.Date.astype(str).str.slice(5,7)
        grouped = sub_data.groupby('month')
        for key, group in grouped:
          i+=1
          monthly_agg_data.loc[i, "station"] =station
          monthly_agg_data.loc[i, "month"] = key
          monthly_agg_data.loc[i, "rain_mm"] = group.rain_mm.mean()
          print(monthly_agg_data.station)


      import calendar
      monthly_agg_data['month_name'] = monthly_agg_data['month'].astype(int).apply(lam
  bda x: calendar.month_abbr[x])


      stations = pd.read_csv(stations_filepath)
      stations = gpd.GeoDataFrame(stations)
      stations = stations.iloc[:,:5]
      stations_list = [Point(x, y) for x,y in zip(stations.x, stations.y)]
      stations_list[0]
      stations['geometry'] = stations_list
      stations.plot()
      print(stations.crs)
      stations.crs = {'init' :'epsg:32737'}


      def plot_station(dataframe, column, map_title, legend_title,cmap, output_fp):
        minx, miny, maxx, maxy =  buildings_rain_aggr.total_bounds
        fig, ax = plt.subplots(figsize  = (7, 7))
        stations_proj = stations.to_crs(epsg=3857)
```

```
        map_plot = stations_proj.plot(ax =ax,figsize=fig, column=column, s=100, alpha=1, legend = True)
        ax.grid(b=True, which='minor', color='#D3D3D3', linestyle='-')
        ax.set_aspect('equal')
        map_plot.text(x=minx+3000,y=maxy-4000, s=u'N \n\u25B2 ', ha='center', fontsize=37, weight='bold', family='Courier new', rotation = 0)
        ax.get_legend().set_bbox_to_anchor((1, 0.44))
        ax.get_legend().set_title(legend_title)
        ax.get_figure()
        ax.set_aspect('equal')
        plt.xlim(minx-1000, maxx+1000)
        ax.set_title(map_title, fontsize=15)
        plt.savefig(output_fp, dpi=300, bbox_inches='tight', pad_inches=0.1)

    cmap='Blues'
    map_title=' in Taita Region'
    legend_title='Weather Stations'
    output_fp = r'E:\LIDAR_FINAL\data\plots\weather_stations_5'
    plot_station(stations, 'Location', map_title, legend_title,cmap, output_fp)

    fig, ax = plt.subplots(figsize  = (9, 5))
    buildings_rain_aggr.plot(ax=ax)
    stations.plot(ax=ax, c='red', column='Location', legend=True)
    plt.xlim(400000, 480000)
    plt.ylim(9610000, 9645000)

    buildings_rain_aggr.crs = stations.crs
    len(stations)
    ground_stations_rain_model = gpd.sjoin(stations, buildings_rain_aggr, how='inner', op='intersects')
    ground_stations_rain_model.Location

    stations_abbr = [station.split(',')[0].split(' ')[0] for station in ground_stations_rain_model.Location]
    wundayi_index = stations_abbr.index('Wundanyi')
    stations_abbr[wundayi_index] = 'Taita_RS'

    ground_stations_rain_model['station'] = stations_abbr

    def get_column_names_lists(ending, except_this):
      return [month for month in ground_stations_rain_model.columns if month.endswith(ending) and not month.startswith(except_this)]
    months_rain = get_column_names_lists('rain', 'ann')
    months_rain_pot = get_column_names_lists('PO', "ann")

    stations_rain_model_df = pd.DataFrame(columns=[ 'station', 'month', 'model_rain_mm','rain_pot'])
    for i, row in ground_stations_rain_model.iterrows():
      monthly_rain = row[months_rain]
      months_list = [month[0:3] for month in monthly_rain.index]
      monthly_rain_pot = row[months_rain_pot].tolist()
      month, rain, rain_pot = monthly_rain.index, monthly_rain, monthly_rain_pot
```

```
        data = {'station': row.station, 'month': months_list, 'model_rain_mm': monthly_rain, 'rain_pot':rain_pot, 'x':row.x,'y':row.y,'z':row.z}
        each_station = pd.DataFrame(data)
        stations_rain_model_df = pd.concat([each_station, stations_rain_model_df], sort=False)


    stations_rain_model_df.columns
    monthly_agg_data.columns
    joined = pd.merge(stations_rain_model_df, monthly_agg_data, left_on=['station','month'], right_on=['station', 'month_name'])
    joined.columns
    joined['rain_err'] = joined['rain_mm']- joined['model_rain_mm']
    import numpy as np

    def rmse(predictions, targets):
        return np.sqrt(((predictions - targets) ** 2).mean())

    rmse_val = rmse(joined.model_rain_mm, joined.rain_mm)
    print("rms error is: " + str(rmse_val))


    rain_stat = linregress(joined.model_rain_mm.tolist(), joined.rain_mm.tolist())
    r2 = rain_stat.rvalue**2
    measured_rain, modelled_rain = joined.rain_mm.tolist() , joined.model_rain_mm.tolist()


    def r2(x, y):
        return stats.pearsonr(x, y)[0] ** 2
    print(r2(measured_rain, modelled_rain))
    x=np.array(measured_rain)
    y=np.array(modelled_rain)

    x = pd.Series(measured_rain, name="measured rain (mm)")
    y = pd.Series(modelled_rain, name="modelled rain (mm)")
    ax = sns.jointplot(x, y, kind="reg", stat_func=r2, logx=True, truncate=True, space=0.1)
    plt.subplots_adjust(top=0.9)
    ax.fig.suptitle('Modelled Rainfall vs Measured Rainfall', fontsize=20)
    # can also get the figure from plt.gcf()
    output_fp = r'E:\LIDAR_FINAL\data\plots\validation_modelled_measured3'
    plt.savefig(output_fp,  bbox_inches='tight',dpi=300, pad_inches=0.1)


    sns.set(color_codes=True)
    discontinued_station = 'Mwatate'
    if discontinued_station in stations_names_list:
      stations_names_list.remove(discontinued_station)
```

```
236.
237.       #plt.rcParams.update({'font.size': 17})
238.       min_rain, max_rain = joined.rain_mm.min()-20, joined.rain_mm.max() + 30
239.       fig, axes = plt.subplots(3, 2, figsize=(10,12), sharex=True, sharey=True)
240.       for i, (ax, station) in enumerate(zip(axes.flatten(), stations_names_list), 1):

241.         sub_data = joined.loc[joined['station']==station].sort_values(by='month_y')
242.         ax.plot(sub_data.month_name, sub_data.rain_mm, lw = 2, color = 'blue', label=
    'Measured')
243.         ax.plot(sub_data.month_name, sub_data.model_rain_mm, lw = 2, color='red', labe
    l= 'Modelled')
244.         ax.legend()
245.         # Figure title
246.         fig.suptitle('Comparison of Measured and Modelled Mean Monthly\nRainfall in Ta
    ita Region', fontsize=23)
247.         station = rename_stations(station)
248.         design_multi_plots(ax, station, min_temp, max_temp)
249.         plt.setp(ax.xaxis.get_majorticklabels(), rotation=90)
250.         plt.tight_layout()
251.       plt.subplots_adjust(top=0.89)
252.       output_fp = r'E:\LIDAR_FINAL\data\plots\monthly_validation_modelled_measured6'
253.       plt.savefig(output_fp,  bbox_inches='tight',dpi=300, pad_inches=0.1)
254.
255.
```

Survey Form

QUESTIONNAIRE FOR RAINWATER HARVESTING SYSTEM

Sample Number/House Number_____

Age: ____ Gender: Occupation:_____ Location:_____

BACKGROUND INFORMATION

1. Number of people in household: _____
2. Income level a)Low b) Medium c) High
3. Education level: a) Primary b) Secondary(O'level) c)Diploma d)Vocational e)University) f)No formal education

COLLECTING RAINWATER

1. Current source(s) of water: (a) Well (b.) Borehole (c.)Roof Water Harvested rain (d) dam (e) river (f) tap water from Taita County by Govt (g)Tap water from Community Project (i)others_____

2. Details of water demand:

i) Average daily water consumption (20 Litres gallon) :

_____

ii) Cost per Month:_____

3. Rainwater Harvesting:

1. Do you have a RWH system? a)Yes (b) No
2. How much did it cost you?

Tank's size : _____

→ yes

1. Do you harvest all the water? a) Yes b) No
2. Uses?
3. How much water you get from collecting?

→ NO : If not, why don't you have one?

a) Too expensive b) Lack of technical expertise c) We don't find it useful d) We don't need it e) House Design. (f)House not Mine g) Health Reason (i) others

WATER CONSUMPTION

1. Do you ever face water shortage? a) Yes b) No.

1a) How often? b) Duration

2. How long do you travel to get water?
3. How long do you queue/Wait?

AFFORDABILITY

6. Do you think it will be nice to have a system to help you save rainwater which can be used at home for domestic use? a) Yes b) No
7. Would you like to install the rainwater harvesting system? a) Yes b) No

8. a) If Yes, Why would you?

_____

_____

b) Preferred tank capacity? _____

9. If No, Why would you not?

_____

_____

ROOF

1. **Roof type/material**
   a) Galvanised Iron Sheet b) Asbestos c) concrete roof d)Grass thatched e)
   Others_____
2. Why this material?

Thank you for responding!

FIELD OBSERVATION FOR THE RAINWATER HARVESTING SYSTEM

Sample Number/House Number_____

House Location(long/lat):_____

Building Type

a) Single family home
b) Residential apartment/ block
c) Commercial building
d) Hotel
e) School
f) Other____

Roof's Condition a) Good b)Fair c) Poor

Rainwater Harvesting
Roof Area (sqm) ____
Tank Storage Size _____

Tank Storage Location:
a) Indoor b) Underground c)Above ground d) Outdoor
e)Others_____
__