 doi:10.20944/preprints202301.0558.v1

Article

Extreme Rainfall Events Detection Using Machine Learning for Kikuletwa River Floods

Lawrence Mdegela [1,*], Esteban Municio [2], Yorick De Bock [1], Edith Luhanga [3] and Steven Latré [1]

[1] University of Antwerp - imec; {lawrencenehemiah.mdegela,Yorick.DeBock,Steven.Latre}@uantwerpen.be
[2] i2CAT Foundation; esteban.municio@i2cat.net
[3] Carnegie Mellon University Africa; eluhanga@andrew.cmu.edu
* Correspondence: lawrencenehemiah.mdegela@uantwerpen.be; Tel.:+32494594736

Abstract: Advancements in Machine Learning techniques, availability of more data-sets, and increased computing power have enabled a significant growth in a number research areas. Predicting, detecting and classifying complex events in earth systems which by nature are difficult to model is one of such areas. In this work, we investigate the application of different machine learning techniques for detecting and identifying extreme rainfall events in a sub-catchment within Pangani River Basin, found in Northern Tanzania. Identification and prediction of extreme rainfall event is a preliminary crucial task towards success in predicting rainfall-induced river floods. To identify a rain condition in the selected sub-catchment, we use data from five weather stations which have been labeled for the whole sub-catchment. In order to assess which Machine Learning technique suits better for rainfall identification, we apply five different algorithms in a historical dataset for the period of 1979 to 2014. We evaluate the performance of the models in terms of precision and recall, reporting Random Forest and XGBoost as the ones with best overall performance. However, since the class distribution is imbalanced, the generic Multi-layer Perceptron performs best when identifying the heavy rainfall events, which are eventually the main cause of rainfall-induced river floods in the Pangani River Basin.

Keywords: Heavy rainfall; River floods; Machine learning ;

1. Introduction

Rainfall-induced river floods are among Earth's most common and most catastrophic natural hazards. According to the Tanzania Meteorological Agency, in the last decade, the northern part of the country has experienced its heaviest rainfall accompanied by strong winds, causing the most severe floods of the last 50 years. It is without a doubt that, with the changing climate, such events are likely to become more frequent, not only in Tanzania but across the globe. The effects of floods are notably severe in developing or low-income countries like Tanzania because of their vulnerability to the occurrence of these phenomena. The vulnerability is partly due to limited human capacity and limited resources invested in dealing with the problem.

Worldwide, flash floods account for more than 5000 deaths annually with a mortality rate more than 4 times greater than other types of flooding [1], and subsequently, their social, economic, and environmental impacts are significant.

It has previously been reported that understanding the trends and key patterns in the occurrence of rainfall events, is, without a doubt, an important step towards better flood risk management plans that will help in designing more accurate early warning systems. [2]. Machine Learning (ML) presents the ability to identify the hidden patterns and trends in the historical climate data [3], and may be used predict the key rainfall events that are associated with the occurrence of floods.

In this work we apply different ML techniques to identify and classify rainfall events in the Karanga-Weruweru-Kikafu sub-catchment, located within the Pangani River Basin,

 doi:10.20944/preprints202301.0558.v1

Tanzania. We compare these techniques and discuss the suitability of each in successfully classifying rainfall events.

In order to train the models, we use an in-field gathered and labeled data-set from five stations located across the Karanga-Weruweru-Kikafu sub-catchment. The nature of the data-set gives us an imbalanced multi-class classification problem. Imbalanced classification is when the distribution of examples in the training data-set across the classes is not equal.

There are three categories in the target class (Heavy, Light and No-rain). Of these, heavy rainfall is the smallest making up just 0.32% of the whole data set. The distribution is highly skewed towards the majority class, in this case, light rain, which make up 83.22% of the whole data set leaving 16.46% to no-rain class. This simply means for every single example of heavy rainfall event, there are 51 examples of no-rain and 260 examples of light rain. In this sense, the contribution of this work are twofold:

* First, we compare different ML techniques in order to assess which is the best approach for rainfall classification. We discuss the results and provide an overview of which methods are better for which scenarios and why.
* Secondly, we present and describe our data set, making it publicly available for future research.

The remainder of this article is organized as follows:

Section 2 presents some relevant background on related works and usage of ML for the purpose of rainfall classification.

Section 3 describes in detail the Karanga-Weruweru-Kikafu sub-catchment weather data set and details the labeling process. Section 4 presents results of the ML techniques applied and highlights the suitability of each for the given problem. Finally, Section V concludes this paper.

2. Background on used methods

In this section we briefly introduce the basic concepts of the algorithms selected to detect rainfall events. The algorithms were chosen based on their ability to deal with nonlinear data. Nonlinear algorithms are drawn from the field of machine learning and make assumptions about the functional form of the problem. They are nonlinear because the output is often a nonlinear mapping of inputs to outputs.

2.1. Random Forest

Random forest is a supervised machine learning algorithm that is used widely in classification and regression problems. The core unit of a random forest is the decision tree. It is a hierarchical structure (see Figure 1) that is built using the features of the data set [4]. Random Forest is based on the ensemble technique and it tends to give accurate results even without parameter tuning [5]. A random forest is a meta estimator that fits a number of decision tree classifier on various sub-samples of the data-set and uses the average to improve the predictive accuracy and to control over-fitting. The sub-sample is controlled with the $max_{samples}$ parameter if $bootstrap = True$ (default), otherwise the whole data-set is used to build each tree. Due to it's capability to deal with nonlinear data, which is usually the case in many real world problems, Random forest has been applied in a number of studies [6] , [7]. Furthermore random forest is also capable of dealing quite well with imbalanced data-set as observed by [8] and [9].

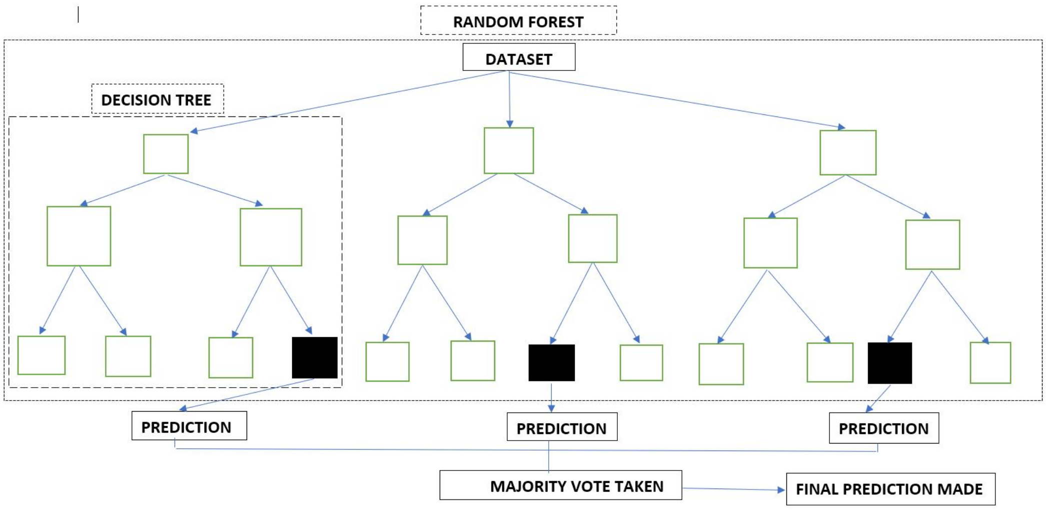


Figure 1. General architecture of Random Forest

2.2. XGBoost

XGBoost stands for eXtreme Gradient Boosting, is an implementation of gradient boosted decision trees designed for speed and performance. Boosting is an ensemble technique where new models are added to correct the errors made by existing models. In this technique, models are added sequentially until no further improvements can be made [10]. Gradient boosting uses a gradient descent algorithm to minimize the loss when adding new models. It has been reported by [11] that XGBoost can perform better than other methods on an imbalanced data-set.

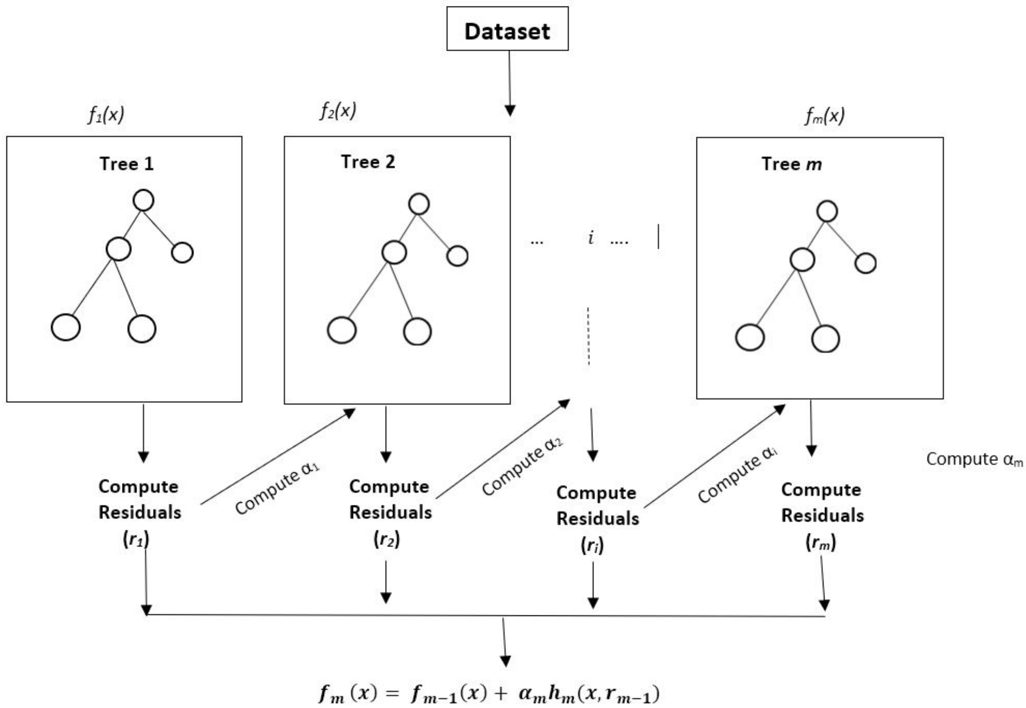


Figure 2. A general architecture of XGBoost

From the general architecture (see Figure 2) of XGBoost, α_i and r_i are the calibration parameters and the residuals (weakness) of the previous model computed with the i^th tree

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2023 doi:10.20944/preprints202301.0558.v1

respectively. h_i is a function trained for predicting residuals, r_i using x for the i^th tree. In order to compute, α_i we use the residuals computed, r_i and come up with the following:

$$\arg\min_{\alpha} = \sum_{i=1}^{m} L(y_i, f_{i-1}(x_i) + \alpha h_i(x_i, r_{i-1})) \tag{1}$$

where $L(y, f(x))$ is a differentiable loss function.

2.3. Support Vector Machine

Support Vector Machine (SVM) is a supervised kernel machine learning algorithm used for both classification and regression. SVM fall under a class of machine learning algorithms called kernel methods in which features can be transformed using kernel functions. This makes them efficient in dealing with any nonlinear pattern in a data-set [12]. Although primarily designed for binary classification, it has been used in several studies to also perform multi-class classification [13–15]. The objective of the SVM is to separate a given data-set the best way possible. After separation, the distance between the nearest points is what is known as *margin*, followed by selecting a *hyperplane* with the maximum possible margin between the *support-vectors* (plus and minus) in the given data-set (see Figure 3).

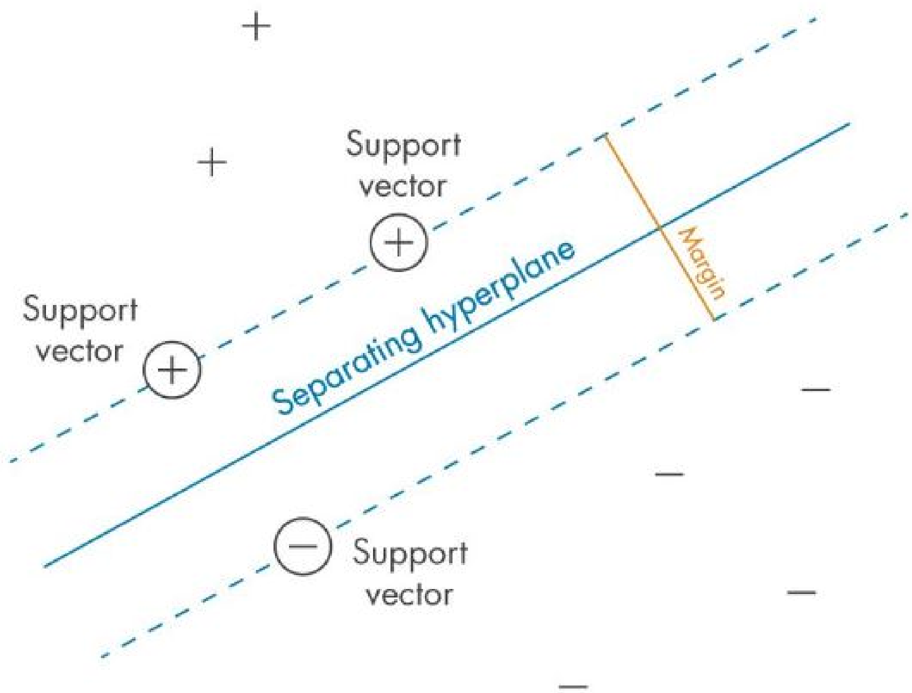


Figure 3. Defining the *margin* between classes – the criterion that SVMs seek to optimize. source: MathWorks[1]

2.4. k-Nearest Neighbors

k-Nearest neighbors algorithm (kNN), is a distribution free, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point [16]. While it can be used for either classification or regression problems, it is mostly applied in classification problems working on the assumption that similar points can be found near one another. kNN is considered one of the simplest machine learning algorithms, however it has been successful in a number of applications, from recognition of handwritten texts [17] to satellite image scenes [18] . Being a distribution-free algorithm, it has shown success in classification problems with irregular decision boundary.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2023 doi:10.20944/preprints202301.0558.v1

Essentially, in kNN, for a given value of K, the algorithm will find the K nearest neighbor of unseen data points, by assigning the class to these unseen data points with the class which has the highest number of data points out of all classes of K neighbors.

2.5. Multi-Layer Perceptron

Multilayer Perceptron (MLP) is a fully connected type of feed forward neural network made of three types of layers- the input layer, output layer and hidden layer as in Figure 4. In each of the layers, there is at least one neuron where the input vector is processed by the MLP in a forward direction passing through each of the layers [19]. The inputs are combined with initial weights in a weighted sum and subjected to the activation function with each combination being propagated to the next layer. This goes all the way though the hidden layer to the output layer. The results, however doesn't end there otherwise there would be no significant learning into minimizing the cost function and that is where back-propagation takes on. The idea of back-propagation of MLP is to adjust, in iterations, the weights in the network in order to minimize the cost function.

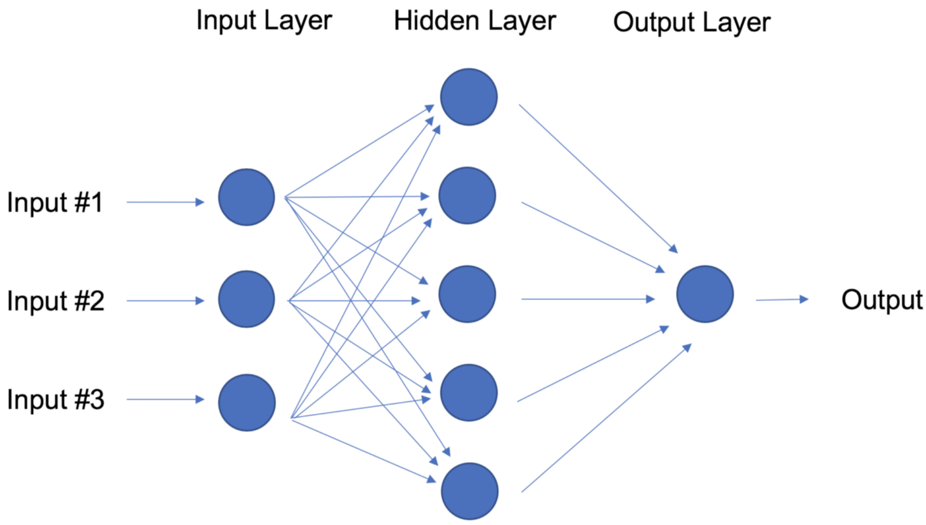


Figure 4. A neural network with a single hidden layer

3. Data sets and Methodology

The study area under consideration is situated in the Northern part of Tanzania in the south of Kilimanjaro region. The Karanga-Weruweru-Kikafu (KWK) sub-catchment (Figure 5) and the villages along the Kikuletwa river are intensely affected by flash river floods from heavy rainfall. The aim of this work was to predict rainfall intensity (heavy, light, none) based on 6 weather parameters namely temperature (Minimum and Maximum), relative humidity, precipitation, solar, and wind speed. Data on precipitation, temperature, relative humidity, solar and wind speed for the catchment area from 1979 to 2014 was collected from the Tanzania Meteorological Agency [2], which uses satellites for measurement. This was complemented my rainfall data from ground gauge stations under the Pangani Basin Water Board(PBWB) [3].

Heavy rainfall prediction play an important role in establishing some thresholds that can be used as input in predicting river floods. Original data set was collected from 5 different stations inside the KWK sub-catchment. The data contains daily weather data for 35 years, with 7 parameters namely Maximum and Minimum temperatures in centigrade(C), precipitation in millilitres(mm), wind speed in meter per second(m/s), relative humidity a fraction, solar in mega-joules per square metre(MJ/m^2) and rain category. Rain gauges located in the study area were used to record rainfall estimates

[2] https://www.meteo.go.tz/
[3] https://www.panganibasin.go.tz/

 doi:10.20944/preprints202301.0558.v1

from the ground, however since they cannot cover a large enough area, it is not easy to do prediction based only on the few local ground gauges. In this sense, the data set is complemented with radar estimates which cover a relatively larger area.

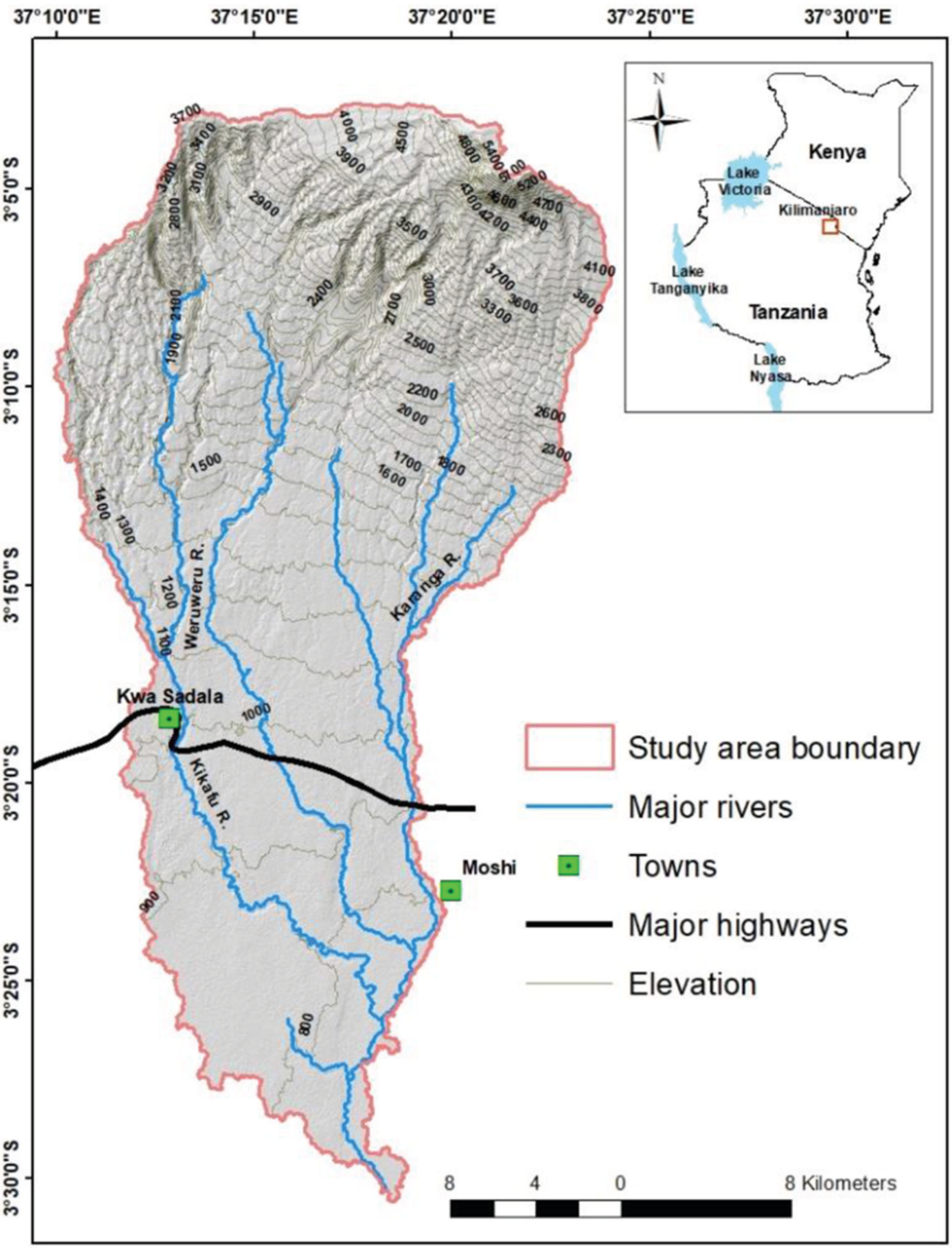


Figure 5. Karanga-Weruweru-Kikafu(KWK) sub-catchment.

3.1. Data preparation

The data from each station were checked for missing values before being merged into a single data set. Simple line plots were used to check whether all stations had similar patterns in the features and to identify any outliers.

We further investigated our target class distribution by using a counter method in python to check for any imbalance. We then conveniently split the data set into train and test set in the ratio of 80% to 20%, in order to even out the distribution as there is an imbalance in the target class distribution, we used stratified sampling. In this context stratification

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2023 doi:10.20944/preprints202301.0558.v1

simply means that a *train_test_split* method returns training and test subsets that have the same proportions of the class labels and the input data set. In total there are 6 features and 1 target.The features are maximum temperature, minimum temperature, precipitation, wind, relative humidity and solar, while the target is rainfall category. (Table 1). Further feature engineering was done, where all the object type columns were encoded to numeric type. Pivoting was also done to put the data in a format that is convenient for model training. Training data was also normalized using *MinMaxScaler* from sklearn library. .

Max.Temp	Min.Temp	Precipitation	Wind	Rel. Humid.	Solar	Category
27.61	11.65	0.00	2.49	0.65	19.68	N
21.39	13.71	3.10	2.70	0.74	11.01	L
23.79	8.85	0.00	2.564	0.641	25.780	N
24.75	8.67	0.00	2.25	0.58	20.04	N
27.739	15.405	0.235	2.598	0.590	19.709	L

Table 1. First few rows of the original data-set

3.2. Model Building

Two main things were considered during this stage before jumping into the models. First the target class distribution and second the multi-class classification. Our target class was somehow severely imbalanced, with the distribution being highly skewed towards the majority class, light rain(83.22%), followed by no-rain (16.46%),and heavy rainfall, which is our class of interest, is the minority (0.32%). This simply means for every single example of heavy rainfall, we have 51 examples of no-rain and 260 examples of light rain.

Consideration on the part of multi-class classification was to use a multi-class strategy from scikit-learn [4] library known as One-vs-the-rest(OvR). OvR is a heuristic technique of dealing with multi-class problems by fitting one classifier per class. For each classifier, the class is fitted against all the other classes. One of the implementation of OvR is from the sklearn library, which provides a separate *OneVsRestClassifier* class that allows the one-vs-rest strategy to be used with any classifier. A classifier that is inherently for binary classification is just provided to the *OneVsRestClassifier* as an argument. Each model was then trained, tested and evaluated. Since our problem fall under multi-class imbalanced classification, selecting a metric for evaluation was the most important step in the project. A wrong metric would mean choosing the wrong algorithm, consequently solving a different problem from the one you wan to solve.

3.3. Model Evaluation

Since we are dealing with a highly skewed data-set we chose Precision and Recall as our performance evaluation metric. Precision(eq:2) is a ratio of the number of true positives divided by the sum of the true positives and false negatives. In other words, it inform about how good a model is at predicting the positive class.

$$Precision = \frac{TruePositives}{TruePositives + FalsePositives} \tag{2}$$

Recall (eq: 3) on the other hand is the ratio of the number of true positives divided by the sum of the true positives and the false negatives.

$$Recall = \frac{TruePositives}{TruePositives + Falsenegatives} \tag{3}$$

One important aspect of precision and recall to take note is that, the calculations do not consider the use of the $truenegatives$. The focus is on the correct prediction of the minority

[4] https://scikit-learn.org/stable/modules/generated/sklearn.multiclass OneVsRestClassifier.html

 doi:10.20944/preprints202301.0558.v1

class. Precision-recall curve is a plot of the precision on the y-axis and the recall on the x-axis for different thresholds. They give a more informative picture of an algorithm in skewed data-sets as it has also been evident in a number of studies [20], [21]. In that sense we identified our positive class to be H for heavy rainfall and other collectively as negative classes(No rain and Light rain). Nevertheless, precision and recall, are in a trade-off relationship, at some point you may need to optimize one at the expense of the other [22]. Contextually, at some point you would want classifier that is good at minimizing both the false positives and false negatives, meaning, it would make more sense to have a model that is equally good at identifying cases were a false alarm of a heavy rainfall event goes off and when an alarm is not going-off while there is an event coming. In the view of that, we applied another metric called F1-score. F1score is the harmonic mean (eq.4) of precision and recall and ranges from 0 to 1.

$$F1 - score = \frac{2 * (precision * recall)}{precision + recall} \tag{4}$$

4. Results and Discussion

In our experiments, five different machine learning algorithms were used to predict extreme rainfall events among three rainfall categories. As stated in the introduction, the ability to identify extreme rainfall events is crucial for predicting rainfall-induced river floods. Results from evaluation show that, overall Random Forest and XGBoost performed better than the rest as we can see from the F-scores summarized in Table 2.

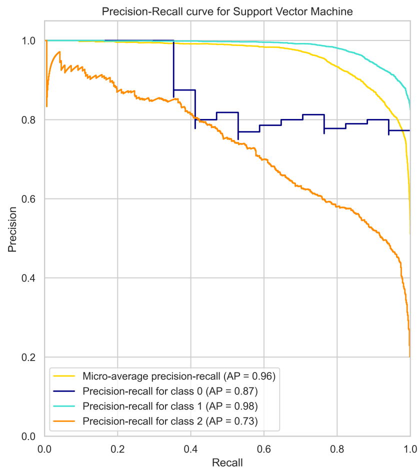
Table 2. Summary of F-score measures for the models

Random Forest	XGBoost	Support Vector Machine	KNN	Multi-layer Perceptron
0.998	0.998	0.878	0.898	0.950

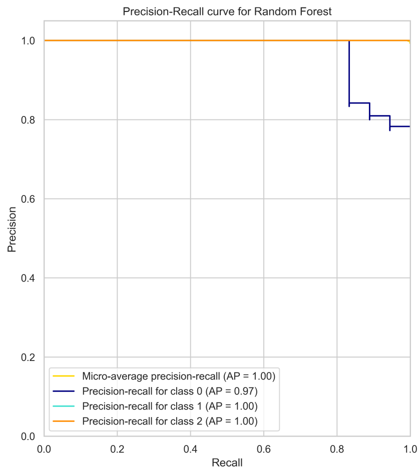
Ideally, the scores from F1-score means both XGBoost and Random forest have perfect precision and recall when you give equal importance to both false negatives and false positives. However, that is not the case in our problem. In classifying the heavy rains, which in our case is the minority class, false negatives were the most important. Intuitively, in our context, it is not helpful if we are successful in predicting all data points as negative, that is no heavy rainfall event, instead, we focused on identifying the positive cases, the occurrence of a heavy rainfall event. Coming back to the metrics language this simply means we maximized the *recall*, the ability of our model to find all the relevant cases within a given data-set. This notion is supported by a number of past studies [23], [24], [25], [26]. In the view of that, F-score is not the determinant for the appropriate model to use in this scenario, as we said it being the harmonic mean, it takes into account both the precision and recall. Our main goal is favor the minimization of the false negatives and not to cast equal importance to both the false negatives and false positives. We focused on having a model with high recall which is able to identify most of the heavy rainfall events(true positives), that way saving lives and properties from the consequences that come along with such events. On the other hand of course, that is at the expense of issuing false alarms of heavy rainfall events as though they will happen (false positive) but they won't. Potentially, the associated costs of false positives will be the unnecessary anxiety to the people and at the worst, the costs associated with taking unnecessary precautions. In most cases, the false positives will not be fatal. Therefore, since false negatives will results into fatalities and destruction, we want to have our classification threshold to favor the optimization of recall over precision. This is the point where we turn our attention to the precision-recall curves for more insight. Despite the fact that Random Forest and XGBoost were the general best performers when we put equal weight on both the false negatives and the false positive, it was the generic Multilayer Perceptron that came on top when we focused on the minority

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 30 January 2023 doi:10.20944/preprints202301.0558.v1

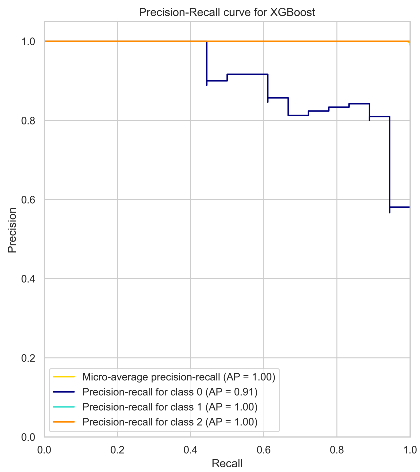
positive class. Multilayer Perceptron (MLP) was 98% in identifying our class of interest as can be seen in the precision-recall curves of figure 6(d).



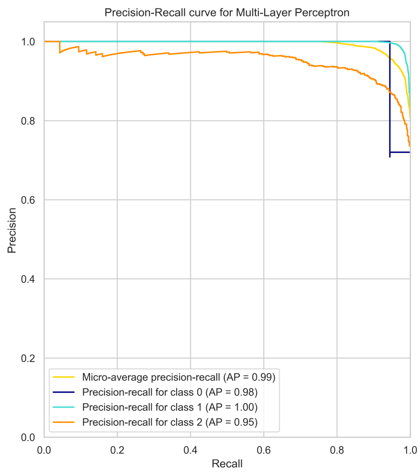
(a)



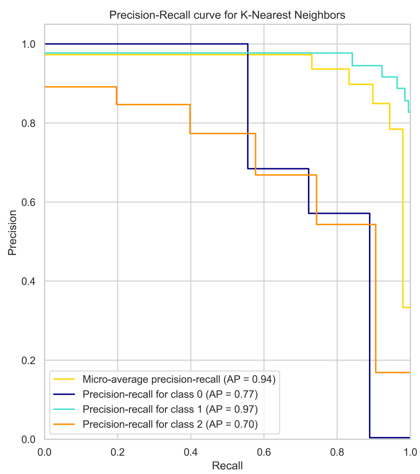
(b)



(c)



(d)



(e)

Figure 6. Precision-Recall Curves for the models

 doi:10.20944/preprints202301.0558.v1

As previously mentioned, precision recall curves give a more informative picture of an algorithm ability in predicting the minority class in skewed data-sets. One explanation to why MLP came on top when it comes into predicting the minority class better than the rest, is the time it took during training. It took relatively longer to train the model, but its predictions were the faster. On the other hand, other algorithms were faster during both training and prediction as compared to MLP. Intuitively this means MLP took time into understanding the hidden patterns in the data-set and that is why it was able to skillfully predict the minority class as compared to the rest of the algorithms. However further study is needed into quantifying the reasons for Multi-layer Perceptron to perform better by looking now at the internal factors such as the hidden layers and other model tuning parameters. Furthermore the results in this study also communicate and solidify the importance of evaluating models based on several metrics in order to pick the hidden skills which can not be seen in just looking at the general accuracy.

5. Conclusion

In this work, we investigated the application of different Machine Learning techniques for detecting and identifying extreme rainfall events in a sub-catchment within Pangani River Basin, found in Northern Tanzania. In order to assess which Machine Learning technique suits better for rainfall identification, we applied five different algorithms in a historical dataset for the period of 1979 to 2014. We evaluated the performance of the models in terms of F-score and precision-recall and we reported Random Forest and XGBoost as the ones with best overall performance. However, since we were dealing with class distribution that was highly imbalanced, the generic Multi-layer Perceptron performed best when identifying the heavy rainfall events, which are eventually the main cause of rainfall-induced river floods in the Pangani River Basin. We specifically showed that reviewing both precision and recall in order to get the best model to predict the minority class in imbalanced classification.

Author Contributions: Conceptualization, L.M and E.M.; methodology, L.M.; software, L.M.; validation, L.M., E.M. and Y.B.; formal analysis, L.M.; investigation, L.M.; resources, L.M and E.M.; data curation, L.M.; writing—original draft preparation, L.M.; writing—review and editing, E.M and Y.B.; visualization, L.M.; supervision, S.L.; project administration, n/a.; funding acquisition, n/a. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Flemish Interuniversity Council for University Development Cooperation (VLIR-UOS), Belgium (Grant number ZIUS2013AP029), through an institutional cooperation programme (IUC) with the Nelson Mandela African Institution of Science and Technology (NM-AIST), under the research project 'Institutional strengthening: ICT, Library and CIC maintenance for collecting, analyzing big data'.

Data Availability Statement: At the moment, data supporting reported results can be found on request, it will be publicly available at the end of the main project

Acknowledgments: The authors would like to thank the Tanzania Meteorological Agency and Pangani River Board for providing some of the data for the study and technical assistance during field activities in the Pangani basin.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

XGBoost	eXtreme Gradient Boost
KNN	k-Nearest Neighbors
ML	Machine Learning
SVM	Support Vector Machine
MLP	Multi-Layer Perceptron
PBWB	Pangani Basin Water Board
TMA	Tanzania Meteorological Agency
KWK	Karanga-Weruweru-Kikavu
OvR	One-vs-the-Rest

References

1. Jonkman, S.N. Global perspectives on loss of human life caused by floods. *Natural hazards* **2005**, *34*, 151–175.
2. Nayak, M.A.; Ghosh, S. Prediction of extreme rainfall event using weather pattern recognition and support vector machine classifier. *Theoretical and applied climatology* **2013**, *114*, 583–603.
3. Parmar, A.; Mistree, K.; Sompura, M. Machine learning techniques for rainfall prediction: A review. In Proceedings of the International Conference on Innovations in information Embedded and Communication Systems, 2017, Vol. 3.
4. Breiman, L. Random forests. *Machine learning* **2001**, *45*, 5–32.
5. Donges, N. Random Forest Algorithm: A Complete Guide, 2021.
6. Luo, C.; Wang, Z.; Wang, S.; Zhang, J.; Yu, J. Locating Facial Landmarks Using Probabilistic Random Forest. *IEEE Signal Processing Letters* **2015**, *22*, 2324–2328. https://doi.org/10.1109/LSP.2015.2480758.
7. Paul, A.; Mukherjee, D.P.; Das, P.; Gangopadhyay, A.; Chintha, A.R.; Kundu, S. Improved Random Forest for Classification. *IEEE Transactions on Image Processing* **2018**, *27*, 4012–4024. https://doi.org/10.1109/TIP.2018.2834830.
8. Khoshgoftaar, T.M.; Golawala, M.; Hulse, J.V. An Empirical Study of Learning from Imbalanced Data Using Random Forest. In Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence(ICTAI 2007), 2007, Vol. 2, pp. 310–317. https://doi.org/10.1109/ICTAI.2007.46.
9. Liu, X.Y.; Wu, J.; Zhou, Z.H. Exploratory Undersampling for Class-Imbalance Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **2009**, *39*, 539–550. https://doi.org/10.1109/TSMCB.2008.2007853.
10. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; Association for Computing Machinery: New York, NY, USA, 2016; KDD '16, p. 785–794. https://doi.org/10.1145/2939672.2939785.
11. Wang, C.; Deng, C.; Wang, S. Imbalance-XGBoost: leveraging weighted and focal losses for binary label-imbalanced classification with XGBoost. *Pattern Recognition Letters* **2020**, *136*, 190–197. https://doi.org/https://doi.org/10.1016/j.patrec.2020.05.035.
12. Pilario, K.E.; Shafiee, M.; Cao, Y.; Lao, L.; Yang, S.H. A Review of Kernel Methods for Feature Extraction in Nonlinear Process Monitoring. *Processes* **2020**, *8*. https://doi.org/10.3390/pr8010024.
13. Bennani, Y.; Benabdeslem, K. Dendogram-based SVM for multi-class classification. *Journal of Computing and Information Technology* **2006**, *14*, 283–289.
14. Mayoraz, E.; Alpaydin, E. Support vector machines for multi-class classification. In Proceedings of the International Work-Conference on Artificial Neural Networks. Springer, 1999, pp. 833–842.
15. Guo, H.; Wang, W. An active learning-based SVM multi-class classification model. *Pattern recognition* **2015**, *48*, 1577–1597.
16. Cheng, D.; Zhang, S.; Deng, Z.; Zhu, Y.; Zong, M. kNN Algorithm with Data-Driven k Value. In Proceedings of the Advanced Data Mining and Applications; Luo, X.; Yu, J.X.; Li, Z., Eds.; Springer International Publishing: Cham, 2014; pp. 499–512.
17. Chychkarov, Y.; Serhiienko, A.; Syrmamiikh, I.; Kargin, A. Handwritten Digits Recognition Using SVM, KNN, RF and Deep Learning Neural Networks. In Proceedings of the CMIS, 2021.
18. Mcroberts, R. A two-step nearest neighbors algorithm using satellite imagery for predicting forest structure within species composition classes. *Remote Sensing of Environment* **2009**, *113*, 532–545. https://doi.org/10.1016/j.rse.2008.10.001.
19. Meyer-Baese, A.; Schmid, V. Chapter 7 - Foundations of Neural Networks. In *Pattern Recognition and Signal Analysis in Medical Imaging (Second Edition)*, Second Edition ed.; Meyer-Baese, A.; Schmid, V., Eds.; Academic Press: Oxford, 2014; pp. 197–243. https://doi.org/https://doi.org/10.1016/B978-0-12-409545-8.00007-8.
20. Cook, J.; Ramadas, V. When to consult precision-recall curves. *The Stata Journal* **2020**, *20*, 131–148, [https://doi.org/10.1177/1536867X2090969 https://doi.org/10.1177/1536867X20909693.
21. Li, W.; Guo, Q. Plotting receiver operating characteristic and precision–recall curves from presence and background data. *Ecology and Evolution* **2021**, *11*, 10192–10206, [https://onlinelibrary.wiley.com/doi/pdf/10.1002/ece3.7826]. https://doi.org/https://doi.org/10.1002/ece3.7826.
22. Erenel, Z.; Altincay, H. Improving the precision-recall trade-off in undersampling-based binary text categorization using unanimity rule. *Neural Computing and Applications* **2012**, *22*. https://doi.org/10.1007/s00521-012-1056-5.
23. Brabec, J., K.; T., F.; V., M. On Model Evaluation Under Non-constant Class Imbalance. In Proceedings of the Computational Science – ICCS 2020; Springer International Publishing: Cham, 2020; pp. 74–87.
24. He, H.; Garcia, E.A. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering* **2009**, *21*, 1263–1284. https://doi.org/10.1109/TKDE.2008.239.

25. Kotsiantis, S.; Kanellopoulos, D.; Pintelas, P.; et al. Handling imbalanced datasets: A review. *GESTS international transactions on computer science and engineering* **2006**, *30*, 25–36.
26. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management* **2009**, *45*, 427–437. https://doi.org/https://doi.org/10.1016/j.ipm.2009.03.002.