International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I5P232
Volume 69 Issue 5, 249-256, May 2021

A Real-Time Flood Detection System Based on Machine Learning Algorithms with Emphasis on Deep Learning

Abdirahman Osman Hashi[#1], Abdullahi Ahmed Abdirahman[#2], Mohamed Abdirahman Elmi[#3], Siti Zaiton Mohd Hashi[*4], Octavio Ernesto Romo Rodriguez[5]

[#]*Faculty member, SIMAD University, Department of Computing, Mogadishu, Somalia*

Department of Artificial intelligence and Big Data, Faculty of Computing, University Malaysia Kelantan Kelantan, Malaysia

Department of Computer Science, Faculty of Informatics, İstanbul Teknik Üniversitesi, İstanbul, Turkey

[1]wadani12727@gmail.com, [2]aaayare@simad.edu.so, [3]m.abdirahman@simad.edu.so

Abstract — *A flood is expressed as water overflowing onto the ground, that usually is dry, or an increase of water that has a significant impact on human life, and it is also declared as one of the most usual natural phenomena, causing severe financial damage to goods and properties, as well as affecting human lives. However, preventing such floods would be useful to the inhabitants in order to get sufficient time to evacuate in the areas that might be susceptible to floods before they happen. Regarding the issue of floods, numerous scholars proposed different solutions, for instance, developing prediction models and building a proper infrastructure. Nevertheless, from an economical perspective, these proposed solutions are inefficient for people in countries like Somalia, for instance. Hence, the main objective of the present research paper is to propose a novel and robust model, which is a real-time flood detection system based on Machine-Learning-algorithms and Deep Learning; Random Forest, Naive Bayes J48, and Convolutional Neural Networks that can detect water level and measure floods with possible humanitarian consequences before they occur. The experimental results of this proposed method will be the solution to forth mentioned problems and conduct research on how it can be easily simulating a novel way that detects water levels using a hybrid model based on Arduino with GSM modems. Based on the analysis, the Random-Forest algorithm outperformed other machine learning models regarding the accuracy compared to the alternative classification methods with 98.7% of accuracy. In contrast, 88.4% and 84.2% were achieved using Naive Bayes and J48, respectively. On the other hand, using a Deep Learning approach achieved 87% of accuracy, showing overall good results on precision and recall. The proposed method has contributed to the field of study by introducing a new way of preventing floods in the field of Artificial Intelligence, data mining, and Deep Learning.*

Keywords — *Machine Learning, Naive Bayes, Random Forest, Artificial Intelligence, Convolutional Neural Network, Data Mining, Natural Language Processing*

I. INTRODUCTION

It's well known that natural disasters cannot be avoided. However, pre-alarming systems and proper management can mitigate their severity and impact. Most of the meteorological departments in developed countries have flood-monitoring cells that may not be appropriately equipped with an intelligent and scalable flood alarming system. On the other hand, other countries may not have that department, including our country, Somalia. In consequence, people living in flood-affected or susceptible areas are dealing with the aftermath of floods every year [1]. In Somalia, the dangerous unexpected floods that occurred in the Baladweyn town of the Hiran region last year had reported over 100,000 people had been displaced [5]. As a consequence of that also, River-flooding has, so far, an impact on an estimated 620,000 people in Somalia [6]. More than around 213,800 of these people have been displaced and fled from their houses, a consequence of the heavy rains that happened in Ethiopia that is received across the country be affected, especially in southern regions that is also Hiran Region is among them, according to the UNHCR-led Protection [6].

Increased rainwater since the beginning of May 2020 has stated a sharp rise in water levels in Jubba and Shabelle rivers as a result, and this might lead to severe flooding in the central and southern regions of Somalia. According to the UNHCR-led, the flood magnitude that occurred in Baladweyn last year is reported as the highest water levels in history that occurred in that region, as well as the whole regions in the country. Moreover, according to data collected by humanitarian partners, the current flood levels exceed 50 years previous periods, as it has affected more than 427,000 people, and of these, nearly 174,000 have fled from their homes as a result of the river flooding that occurred in

Hirshabelle state [5].
According to (CMHC), these floods can occur at any time of the year and are most often caused by heavy rainfall that may happen in Ethiopia that would cause then to raise the level of the Shabelle and Jubba rivers. As a consequence, many people have evacuated and lost their houses. Hence, in the last few years, due to the rapid advance of communication technologies, Global Positioning System known as (GPS) that equipped with wireless devices and GSMs have been broadly deployed in various public and private positions, generating huge amounts of data that could be implemented to measure water levels, locations, and so on, for fleet management [7].
In order to predict & detect floods, Machine learning applications can give valuable solutions to tackle this phenomenon case. Moreover, it is another inevitable job to resist the devastation's flood if there is a possible method to inform the population living around the area through the appropriate and proper way in real-time[1][12]. To date, detecting variations of water levels in a variety of flood zones is widely utilized as sensor technologies to share data with inhabitants [13]. The purpose of this paper is to simulate a real-time Flood Detection System based on Machine Learning that can detect water levels and measure floods with possible humanitarian consequences before they occur.
To ensure the safety of people from possible floods, we will implement various Machine learning methods that will verify the results. This will help people have the opportunity to flee areas where flooding is likely to occur [3]. It will also help reduce the impact of the floods on human life and the economy, such as evacuating the movable things that can take people from the flood zone. Hence to verify this, we have chosen the proposed model to apply four machine learning algorithms: Naive Bayes, Random Forest, J48, and Convolutional Neural Network [16]. Naïve Bayes is one of the machine learning models that is based on conditional statistics and used to classify different categories of the data by using it is own classifier. The reason why it was chosen is because of the data that will come from the Arduino system and will be needed to categorize up to three levels (normal, alert, and dangerous) so that a machine learning algorithm was needed to classify these categories, having high accuracy of classification according to the current state of the water [2].
Similarly, Random Forest is one of the machine learning models that are good at handling a large volume of datasets with high dimensionality. It is also capable of performing classification and regression. Given this reason, it was chosen as the classifier for this model [4]. Random Forest uses the concept which is based on decision tree by classifying dataset and predictions that comes from each tree may have a low correlation for certain times. However, it is good with having a good accuracy, and it also prevents the overfitting of the data. On the other hand, J48 is also another machine learning algorithm that is good for the categorization of the data. It is also based decision tree but depletes the performance and accuracy [20]. Furthermore, the addition of a Deep Learning approach by using a CNN to predict floods based on the data gathered by the Flood Detection System extends the scope of the research in more modern fields of Artificial Intelligence, providing an opportunity of developing the research using a variety of techniques and approaches that could benefit its results [19]. This paper is structured with five sections. The following section provides background and related work on flood detection methods. The third section describes the methodology in which this framework is to be implemented, as well as presenting the experiment design of the proposed framework. Finally, the fourth section presents a result analyze, and its conclusion.

II. BACKGROUND STUDY

There are many natural disasters around the globe; however, floods are known to be one of the most critical, causing huge damage to human life, infrastructure, and agriculture [5]. Hence there must the use of some sort of machine learning algorithm. Machine learning is one of the prominent fields in artificial intelligence that came from the improvement of self-learning algorithms to get knowledge from that data so as to create forecasts. These days, the data are huge, and these data can be converted into knowledge by using an algorithm that is the field of machine learning [8].
Machine learning gives a good effective option for taking the knowledge into data to increasingly raise the forecast models' performance and create decisions that came from that data. Hence, the meaning of this research is if we desire to forecast the level of the river in a particular place, we can use a special ML algorithm with our past data, and if it is successfully recognized, then it will do better prediction for future water levels [12]. Artificial-neural networks, neuro-fuzzy are among the numerous ML algorithms that were stated as effective in terms of short and long for flood prediction, and the following subsequence explains each of these algorithms [13].

A. Artificial Neural Networks (ANNs)

Artificial neural networks are systems that have a numerical model with a successful proficient parallel processing. Enabling them to imitate the utilization between neural units and the biological neural network [8]. Among all ML-methods, ANNs are the most important popular learning algorithm, known to be easily changed and effective in modeling complex flood processes, and it has tolerance with a high fault also brings an accurate approximation [9]. If we compare the convention statistical model to ANNs, the ANN approach was utilized with greater accuracy for the help of predictions. Since their first-time usage of ANN In the 1990s, this algorithm is the essential prevalent method for flood prediction [10].

B. Adaptive-Neuro-Fuzzy Inference System (ANFIS)

The fuzzy logic proposed by Zadeh Baydargil [7] mentioned could be some soft computing technique with a qualitatively model technique using natural language. It is also known that Fuzzy-logic is a basic mathematical model for calculation, which works on consolidating expert knowledge into a fuzzy-inference system for the able classification of a different date. An FIS another play actor for human learning through a prediction function with less complexity of computations, which gives the good ability for nonlinear-modeling of extreme hydrological events [9], especially flood ones.

C. Decision Tree (DT) & Ensemble-Prediction-System (EPSs)

The machine language strategy of a decision tree is one of modeling predicative for suppliers with a thick application in stimulation-of-flood [11]. The decision tree uses branches from the tree of decisions that are high precision to the leaves that are the target ones. In classification trees (CT), the last factors in a decision tree have a separate set of values, where the leaves stand for classes as labels and the branches on behalf of relationships of feature labels [13][8]. Meanwhile, a lot of languages simulating machine alternatives were showed flood simulating models having a very tough background [12][14]. Hence, there is a developing approach to vary from a single form of prediction to an ensemble of models which is fit for not many applications, cost, dataset [18 [15].

D. Convolutional Neural Networks (CNN)

This is one of the deep learning models that allow many researchers to be applied various fields. As it becomes a research hotspot and outperformed many other machine learning, it is also part of ML and computer vision applications. One of the main factors that CNN made more important or even extremely more popular is by having a strong and constantly increasing computational power in modern computers. CNN is commonly trained in a supervised fashion as it is a feed-forward network with subsampling and alternating convolutional. It is also notable to mention that Deep CNN has been developed to work with 2D data such as images and videos, and it is usually called 2D-CNNs. 2D-CNNs have a good forward neural network that can extract features accurately and then learn the complex objects from big data of labeled data. Although there are different researchers that have proposed different dimensionality of CNN [1] is one of the researchers that has proposed the first ID-CNN in order to deal with sequential data. Whilst is also one of the researchers that has proposed 2D-CNN, and it has gained popularity and this lead that CNN has been applied in different fields, which are included flood detection prediction and other research areas such as biomedical data classification, sentiment analysis, and structural damage detection [21].

III. METHODOLOGY

The research methodology gives a structured overview of the sequences of the following. The overall framework of this research work will be in different phases, either it is hardware or software development. It is known that a successful early forecasting and flood warning system will benefit the population since it acts as the first stage of initial action for the victims in terms of human affecting and infrastructure damages. While SMS is an appropriate alert announcement tool that can distribute the data to the flood victims within a particular area.

Hence, the first phase is to find out and select scholarly information to acquire the adequate knowledge required to carry out this research. The main source of information and knowledge in this phase would be observing the river and gathering data from the riverside community. An example of adequate knowledge is asking the riverside community for the best place in the river that would be good for implementing this proposed architecture. The second phase is going to be the implementation phase. A water level sensor would be put in the river in order to get and send dynamic real-time data to the flood-control- the device for data mining purposes. This sensor device has its own function. It will detect the water level that could be in normal, above normal or dangerous condition. After the data is collected and converted from analog to digital, Machine learning algorithms will be trained to decide if there is a critical condition or not. Random Forest, Naive Bayes, and J48 are the machine algorithms that would do the classification base on their accuracy.

The last Phase will be the Data Sharing Phase. The data that has been gathered will be transmitted to the core control unit (microcontroller PIC). The PIC will obtain the most accurate data in terms of their classifiers. After the high accuracy data is obtained, the data can be monitored and controlled from any place in the region that is available GSM service.

A. Experimental Design

The proposed framework architecture is designed to be a hybrid model based on machine learning algorithms, with hardware devices that will be able to detect water levels. The data collected from the water sensors will be transmitted to the main controller, the PIC Microcontroller.

a) Software*:* In order to develop the proposed system, it was decided to use Java Programming language using Arduino Platform and Weka for data mining. Java is a general-purpose programming language that has a few implementation dependencies available. It is intended to let the application developers write the code once and run it anywhere. Arduino Software was used as our IDE, as it is a cross-platform application that is written in the programming language Java itself. It is used to write and upload programs to Arduino-compatible boards. But also, with the help of 3rd party cores, other vendor development boards.

Regarding the CNN approach, the programming language used was Python, which is the preferred programing language among data scientists. As for the development environment, Jupyter Notebook was chosen due to the tools that it offers to visualize data and results. CNNs are usually used to analyze, classify and manipulate image data. Essentially, the series of processes that they follow to achieve this task are; first, it takes the whole image as input (input matrix). Then, the image is taken by parts (called windows), where the size of the window is determined by the size of the filter. Notice that there can be multiple filters and, therefore, multiple layers. After that, the parameters of the CNN are adjusted using the input data of the window between the Activation and Loss Functions (these are the so-called layers of the CNN). Afterward, the result of the parameters adjustment is set in a vector containing all the relevant information, then this vector (or vectors, depending on the windows used) will be reduced using a pooling technique to summarize the relevant information. Finally, the result of the pooling is combined in a "feature vector" containing all the relevant information of the CNN in a condensed shape that can be used to finish its process.

In the case of this research, a CNN was designed using an approach related to Natural Language Processing (NLP) on the available floods-related data, which is only text. Generally, the procedure of the CNN is similar. Nonetheless, it is important to point out some differences. For example, the input of the CNN is a vector. Since the algorithms used in the CNN are optimized for numerical inputs, the text data is converted using word2vec. Additionally, the simplicity of the input data made pooling unnecessary.

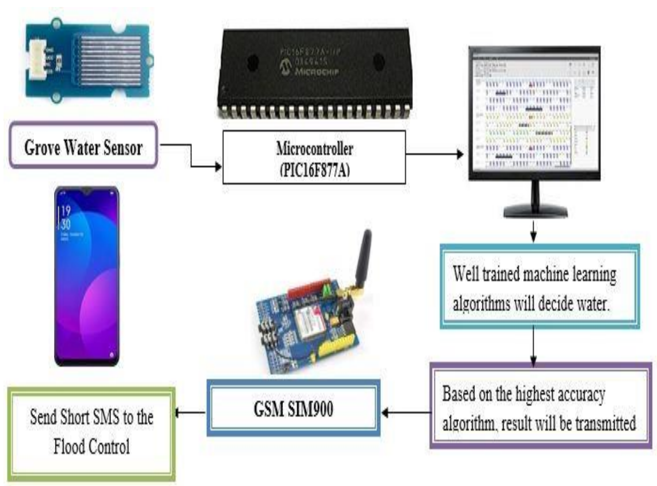

Fig. 1. CNN Overview

***b) Hardware*:** In order to implement the system, a number of hardware devices were used. First, a GSM modem, which is a specialized type of modem that uses a SIM-card, and operates over a subscription to a mobile operator, just like a mobile phone. When a GSM modem is connected to a computer, it allows the computer to use the GSM modem to communicate over the mobile network. In this context, it is mostly used for sending and receiving SMS messages. Secondly, the water sensors that are used are electronic devices that are designed to detect the presence of water for purposes such as providing an alert in time to allow the prevention of water damage. PIC16F877A is an Integrated-Circuit (IC) embedded in a single chip and acts as a voltage level converter. PIC16F877A is capable of converting 5V TTL Logic level to TIA/EIA-232-F level and can take up to +-30V input. It is normally used for the communication between the microcontroller and Laptop/PC.

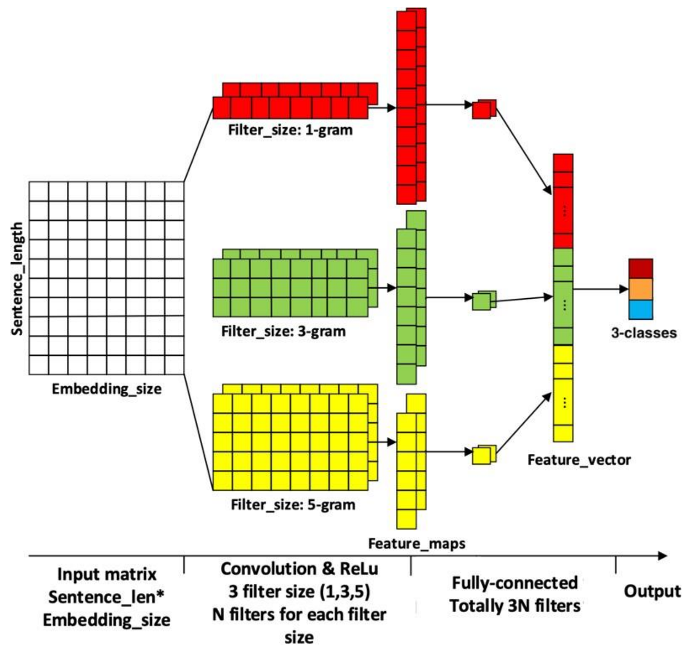

Fig. 2. Proposed Framework for flood Detector

Figure 2 demonstrates the whole process guide for software development and the hardware of the architecture proposed for implementation. The water level sensors were epitomized in order to provide the flood control center with real-time information for processing, intended to be used later. Those sensors have special tasks. They will monitor the water signals, detecting normal levels, while they transfer data to the Microcontroller, and then the machine learning algorithms will decide based on this data. Finally, if there is a dangerous situation, an SMS message will be sent from the GSM SIM900.

IV. RESULT AND DISCUSSION

The anticipated model is based on detecting the water level and training three machine learning algorithms, besides a CNN, to measure the accuracy of the water level. Those are a Random-Forest algorithm, Naive Bayes algorithm, and J48 has implemented. As it was mentioned, generally, there are two key works done in this research: First, using the Arduino and GSM devices to monitor in real-time the river and collecting data as a dataset. While the second step is to mine the collected data and use it in these three selected algorithms in order to know the water level accuracy that is improving the accuracy performance of the flood detection methods.

In the following section, the experiment results and analysis are discussed to contain all the mentioned key components: the Arduino with GSM has successfully fulfilled its essential role in generating a good data collection. With the proper water level parameters setting, it succeeds in achieving better accuracy than an ordinary solution, such as fully relied on sensors.

The upcoming figures are the results that were obtained from the algorithms. Each table or figure is followed by an additional description.

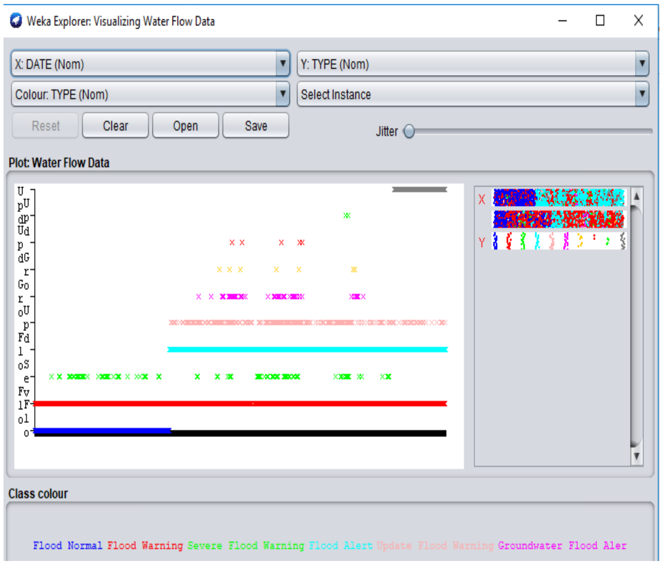

Fig. 3. Normal Water Level

As Figure 3 demonstrates, the water level is normal. However, it is also essential to take into consideration that flood warning is increasing (Red color) while normal water is opposite to that and dramatically decreasing (Blue color). Meanwhile, the three algorithms do not have very different values in terms of correctly classified instances and incorrect classified instances while the water level is normal.

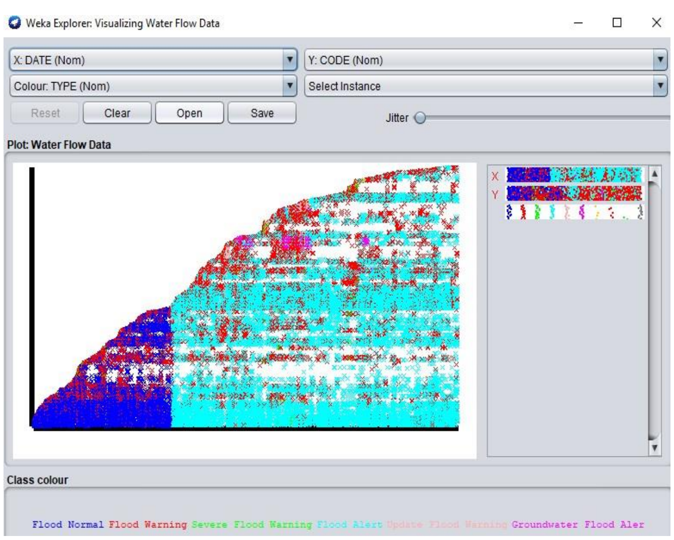

Fig. 4. Flood Water Level

However, as figure 4 shows demonstrate, the water level is increased rapidly; it is possible to observe that the three algorithms have different values in terms of correctly and incorrectly classified instances. Based on the analysis, it can be observed that the three machine algorithms have different variations, and the upcoming table will demonstrate classifier output for their classifications as the upcoming tables will illustrate.

TABLE 1: DETECTED WATER LEVEL IN TERM OF THREE ALGORITHMS

Parameters	Methods		
	Random Forest	**Naive Bayes**	**J48**
Correctly classified instance	98.7%	88.4%	84.2%
Incorrect classified instances	22.8%	2.8%	2.9%
Root mean squared error	0.0904	0.1387	0.1970
Total number of instances	1000	1000	1000

It is important to mention that the three experiments were conducted with imbalanced data, actual data obtained from real flood readings from the Arduino device. Using Random Forest gave as a result of 98.7% of correctly classified instances, whereas the incorrectly classified instances were 22.8%. The Naive Bayes algorithm showed a better result than the J48 algorithm. The correctly classified instances reached 88.4%, while the incorrectly classified ones for this algorithm were 2.8%. On the other hand, the J48 algorithm achieved 84.2% of accuracy for the correctly classified instances, while 2.9% were classified incorrectly. Evidently, Random Forest achieved the best result, compared to the rest of the classification methods, with 98.7 % accurately classified instances. However, 22.8% of the instances classified incorrectly, as mentioned before.

TABLE 2: DETECTED WATER LEVEL IN TERM OF ACCURACY BY CLASS

Method	Accuracy by Class		
	True Positive	**True Negative**	**Recall**
Random Forest	0.989	0.004	0.976
Naive Bayes	0.886	0.014	0.888
J48	0.842	0.021	0.842

Table 2 demonstrates the overall precision of the different classification methods. With the aim of increasing information benefit from the data collected from water levels, a number of machine learning methods were trained to determine the appropriate techniques that would be able to produce a good performance and accuracy. The Random Forest algorithm achieved a better performance than the other methods. Because of that, Random Forest has gotten the highest True Positive, which is 0.989%, whereas J48 obtained the lowest one, which is 0.842%. Meanwhile, in terms of True negative, Random forest achieved the lowest negative, which 0.004%, whereas J48 achieved the highest value for 0.021%. In order to avoid over-fitting issues and generating easy to set constraints, Random Forest can deal with supervised learning algorithms and utilize a huge number of decision-tree models. Using this model will provide help and support to those who are living around the river areas that face many circumstances, such as floods.

Analogously, the CNN approach allowed me to have a better insight into the notion of using Deep Learning techniques in a task such as the one being carried out in this research. Both the development and results of this approach are relevant to understand the impact and potential of this subfield of Artificial Intelligence in similar works.

First of all, as it was briefly mentioned previously, the dataset contains only text data, which makes it unsuitable to use directly as input in a CNN since the algorithms used in it are mostly statistical and probabilistic functions developed to use numerical data. In addition to that, the dataset, although fairly populated, does not contain huge amounts of data to be analyzed; the implications of these characteristics will be discussed in more detail shortly after. The following figure enables us to perceive the simplicity of the dataset, as well as showing the amount of available data (45,141 instances).

```
                        date        code          type
0      01/31/2006 13:55:34    113WACT2a  Flood Normal
1      01/31/2006 16:33:10   112WACTAVN  Flood Normal
2      02/08/2006 11:19:45    054WATBT1  Flood Normal
3      02/08/2006 11:25:26    054WATBT2  Flood Normal
4      02/08/2006 11:32:20    054WATBT3  Flood Normal
...                    ...          ...           ...
45136  06/25/2019 13:39:07    033WAF304   Flood Alert
45137  06/25/2019 13:52:33    033WAF301   Flood Alert
45138  06/25/2019 14:25:34    033WAF207   Flood Alert
45139  06/25/2019 15:39:26    031WAF105   Flood Alert
45140  06/25/2019 18:12:33    034WAF414   Flood Alert

[45141 rows x 3 columns]
Columns in the original dataset:

Index(['date', 'code', 'type'], dtype='object')
```

Fig. 5. Dataset overview

The CNN developed for this research was built to carry out a classification task in order to provide predictions. The target attribute used to perform the classification was the "type" attribute, and therefore the class. There are only *three different classes; "Flood Normal," "Flood Warning," and "Flood Alert." The data is not balanced, and no balancing techniques were applied since it would have resulted in an unrealistic proportion of the classes' representation.

The next figure offers an overview of the class's distribution and the number of instances for each class:

```
Number of rows per type:
Flood Alert      22451
Flood Warning    11658
Flood Normal     11032
Name: type, dtype: int64
```

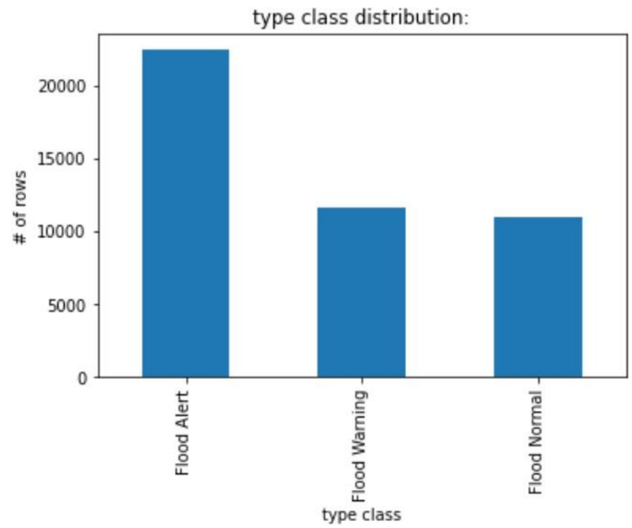

Fig. 6. Class distribution

As mentioned before, the CNN developed was designed using an approach related to Natural Language Processing. In an ordinary CNN for NLP, the original input text data experiences a series of modifications in order to make it more suitable for data analysis. Commonly the text data is tokenized; during this process, the sentences taken as original input are converted to an array of words. This process also involves some degree of data cleaning, removing numerical and non-standard values like numbers, URLs, paths, etc. The other two common steps are Stemming and Lemmatization, which converts the tokens(words) in the input array to their root word in order to simplify the data and remove redundant instances.

Having said that, given the characteristics of the text data in the dataset, none of these techniques was used in the CNN for the research. In fact, several experiments were carried out to analyze the effects of such techniques in the performance of the CNN, and they all showed a negative effect. The reason for that was the particular characteristics of the text data. The text data, due to its simplicity and the fact that it is mostly composed of non-standard words, get an important portion of its data removed when it is tokenized. In a similar manner, when Stemming or Lemmatization is used in the data, a significant part of it is removed since there are mostly non-standard words, there is no root word to link it to, and it is deleted from the input data. Using such techniques had a negative impact between 6% and 12% inaccuracy. Another important aspect of the data and the CNN is that 80% of the data was used for training (36,111 instances) and 20% for testing (9,030 instances).

The following figure offers an overview of the training data, as well as the input vector used by the CNN:

```
       index           type
0       7135  Flood Warning
1      27288    Flood Alert
2      32685  Flood Warning
3       4637  Flood Warning
4      35697    Flood Alert
...      ...            ...
36107  35483    Flood Alert
36108  39296    Flood Alert
36109   2693   Flood Normal
36110   8076   Flood Normal
36111   7624   Flood Normal

[36112 rows x 2 columns]
Epoch1
Epoch ran :1
Epoch2
Epoch ran :2
Epoch3
Epoch ran :3
Epoch4
Epoch ran :4
Epoch5
Epoch ran :5
Input vector
[[ 0  0  6  1  4 10  8  5 34 61 61 61 61 61 61 61 61 61]]
Probs
tensor([[9.9989e-01, 3.9715e-06, 1.0312e-04, 7.6381e-12]],
       grad_fn=<SoftmaxBackward>)
0
```

Fig. 7. Training data and input vector overview

Subsequently, the developed CNN displayed an overall better performance than most of the Machine Learning algorithms, achieving 87% of accuracy while maintaining a good balance in precision and recall. The achieved result is in a sustainable range that gives us the certainty that the model is not overfitted. Nonetheless, it is important to mention some remarks and insights that were gained during the development and while analyzing the results achieved. Firstly, the results achieved proved that using a CNN with a design inspired on NLP is also fit for tasks that require analyzing simple data. Despite that, several adjustments and modifications to the NLP technique had to be done in order to achieve a good performance.

In a similar manner, the simplicity of the data analyzed reduced scientifically the necessity of the number of loops needed to be performed by the CNN. Only five loops were used, and as the next figure shows, between the 4th and 5th loop, the accuracy of the model was very stable, eliminating the need for more loops. This kept the total training time under 5 minutes, which makes the model very convenient for a real-life scenario where the model can be constantly fed and trained to keep a good performance using the most recent data.

Thus, since the model has the capability of being updated in such a short time, it is very likely that the results achieved in this research can be surpassed with more instances of more recent data being used in the CNN.

The following figure shows the accuracy achieved and other helpful measures:

```
Index(['iter', ' loss'], dtype='object')
              precision    recall  f1-score   support

Flood Normal  ---> 0       0.91      0.52      0.66      2212
Flood Warning ---> 1       1.00      1.00      1.00      2318
Flood Alert   ---> 2       0.81      0.98      0.88      4499

    accuracy                           0.87      9029
   macro avg       0.90      0.83      0.85      9029
weighted avg       0.88      0.87      0.86      9029

Index(['iter', ' loss'], dtype='object')
```

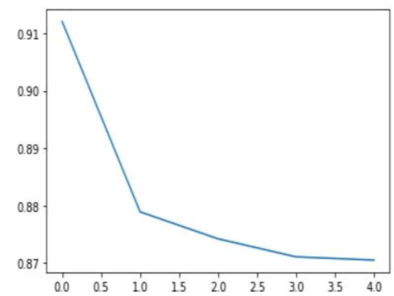

Fig. 8. CNN results

V. CONCLUSION

Systems for flood detection have been developed as an immediate response to inform the corresponding authorities before the event happens. It will keep the authorities updated about the current water levels by means of the Arduino sensor network, which will then provide an SMS notification if there is a dangerous situation through the GSM modem. In order to classify the data, four machine learning methods were used. It was found that the Random Forest algorithm had the best performance regarding accuracy compared to the alternative classification methods with 98.7 % of accuracy, compared to 88.4% using the Naive Bayes algorithm that plays an essential role. Furthermore, J48 achieved 84.2% accuracy close to the Naive Bayes; however, it is slightly lower than that algorithm. With regard to the CNN approach, it achieved 87% of accuracy, which seems lower at first sight, but had a better overall performance in respect of other metrics such as precision or recall. In addition, it was found that the performance of the model was affected negatively when using common NLP techniques. In other words, preserving the simplicity of the data is important in researches such as this one.

However, this proposed method can be further improved or enhanced to achieve to do more advanced technology and well applications that are capable of data mining in the next phase of the research. Subsequent enhancements for this proposed architecture could be advanced by adding video surveillance and tracking the installed equipment using a GPS module. Finally, clustering algorithms could be applied as well in order to improve the results of the proposed model.

ACKNOWLEDGMENT

We, the authors of this paper, would like to extend our gratitude to SIMAD University for funding and supporting Research Grant no. 15. We gratefully acknowledge the

SIMAD Research Center for making this research possible by offering their support.

REFERENCES

[1] Khalaf, M., Hussain, A.J., Al-Jumeily, D., Fergus, P. and Idowu, I.O., 2015, September. Advance flood detection and notification system based on sensor technology and machine learning algorithm. In 2015 International Conference on Systems, Signals and Image Processing (IWSSIP) 105- 108. IEEE.

[2] Kabir, S., Patidar, S., Xia, X., Liang, Q., Neal, J., & Pender, G. (2020). A deep convolutional neural network model for rapid prediction of fluvial flood inundation. Journal of Hydrology, 590, 125481.

[3] Arshad, B., Ogie, R., Barthelemy, J., Pradhan, B., Verstaevel, N., & Perez, P. (2019). Computer vision and iot-based sensors in flood monitoring and mapping: A systematic review. Sensors, 19(22) 5012.

[4] Sankaranarayanan, S., Prabhakar, M., Satish, S., Jain, P., Ramprasad, A., & Krishnan, A. (2020). Flood prediction based on weather parameters using deep learning. Journal of Water and Climate Change, 11(4) 1766-1783.

[5] Halbeeg. (2017, April 27). 100,000 People Displaced by Floods in Beledweyne. Retrieved from https://en.halbeeg.com:https://en.halbeeg.com/2018/04/27/100000-people-displaced-by-floods-in- beledweyneofficial/

[6] OCHA. (2018). Floods: Response plan. Humanitarian Country Team and partners. K. Elissa, "Title of paper if known," unpublished.

[7] Baydargil, H.B., Serdaroglu, S., Park, J.S., Park, K.H. and Shin, H.S., 2018, September. Flood Detection and Control Using Deep Convolutional Encoder-decoder Architecture. In 2018 International Conference on Information and Communication Technology Robotics (ICT-ROBOT) (pp. 1-3). IEEE.

[8] Mosavi, A.; Rabczuk, T.; Varkonyi-Koczy, A.R. Reviewing the novel machine learning tools for materials design. In Recent Advances in Technology Research and Education; Springer: Cham, Switzerland, (2017) 50–58.

[9] Li, L.; Xu, H.; Chen, X.; Simonovic, S. Streamflow forecast and reservoir operation performance assessment under climate change. Water Resour. Manag. 2010, 24, 83.

[10] Wu, C.; Chau, K.-W. Data-driven models for monthly streamflow time series prediction. Eng. Appl. Artif. Intell. 23(2010) 1350–1367.

[11] Zadeh, L.A. Soft computing and fuzzy logic. In Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers by Lotfi a Zadeh; World Scientific: Singapore, (1996) 796–804.

[12] Corbin, B.; Khalighi-Sigaroodi, S.; Malekian, A.; Ahmad, S.; Attard, P. Drought forecasting in a semi-arid watershed using climate signals: A neuro-fuzzy modeling approach. J. Mt. Sci. 11(2014) 1593–1605

[13] Choubin, B.; Khalighi-Sigaroodi, S.; Malekian, A.; Ki¸si, Ö. Multiple linear regression, multi-layer perceptron network, and adaptive neuro-fuzzy inference system for forecasting precipitation based on large-scale climate signals. Hydrol. Sci. J. 61(2016) 1001–1009.

[14] Dineva, A.; Várkonyi-Kóczy, A.R.; Tar, J.K. Fuzzy expert system for automatic wavelet shrinkage procedure selection for noise suppression. In Proceedings of the 2014 IEEE 18thInternational Conference on Intelligent Engineering Systems (INES), Tihany, Hungary, 3–5 July (2014) 163– 168.

[15] Hashi, Abdirahman Osman, et al. A Robust Hybrid Model Based on Kalman-SVM for Bus Arrival Time Prediction. International Conference of Reliable Information and Communication Technology. Springer, Cham, 2019.

[16] Tiwari, M.K.; Chatterjee, C. Development of an accurate and reliable hourly flood forecasting model using wavelet–bootstrap–ANN (WBANN) hybrid approach. J. Hydrol. 394 (2010) 458–470.

[17] Amir Mosavi, K.-w. C. (2018). Review Flood Prediction Using Machine Learning Models. Water 2018, 1-41.

[18] Hameed, Sheila S., et al. Filter-wrapper combination and embedded feature selection for gene expression data. Int. J. Advance Soft Compu. Appl 10.1 (2018).

[19] Sajedi-Hosseini, F.; Malekian, A.; Choubin, B.; Rahmati, O.; Cipullo, S.; Coulon, F.; Pradhan, B. A novel machine learning-based approach for the risk assessment of nitrate groundwater contamination. Sci. Total Environ. 644(2018) 954–962.

[20] Duncan, A., Keedwell, E., Djordjevic, S., & Savic, D. (2013). Machine learning-based early warning system for urban flood management.

[21] Han, Z., Li, X., Xing, Z., Liu, H., & Feng, Z. (2017, September). Learning to predict severity of software vulnerability using only vulnerability description. In 2017 IEEE International conference on software maintenance and evolution (ICSME) 125-136. IEEE.