Weather Data Handlings for Tornado Recognition Using mHGN

Benny Benyamin Nasution, Rahmat Widia Sembiring, Muhammad Syahruddin, Nursiah Mustari, Abdul Rahman Dalimunthe, Nisfan Bahri, Bertha Br Ginting, Zulkifli Lubis

HAL Id: hal-02280323
https://inria.hal.science/hal-02280323

Submitted on 6 Sep 2019

Weather Data Handlings for Tornado Recognition using mHGN

Benny Benyamin Nasution, Rahmat Widia Sembiring, Muhammad Syahruddin,
Nursiah Mustari, Abdul Rahman Dalimunthe, Nisfan Bahri,
Bertha br Ginting, Zulkifli Lubis

Politeknik Negeri Medan, Indonesia,
Jalan Almamater No. 1, Kampus USU, Medan 20155, Indonesia
`benny.nasution@polmed.ac.id`

Abstract. The usage of the mHGN as a pattern recognizer cannot necessarily be used to recognize tornados. Two important issues that need to be solved first are related to data handlings of not-accurately recorded data, and to those of complex weather data. The not-so-appropriate data handlings will produce high false positive and true negative rate of the recognition results. Yet, the latest development of those data handlings has been carried out, and has shown positive and promising results. Such a new approach of data handlings can, therefore, be used to improve the quality and the accuracy of forecasting a tornado. The results taken from a simulated circumstances of a multidimensional pattern recognition have shown, that the tornado can be recognized around 9 hours earlier with 90% of accuracy. However, several improvements related to the data representation within the mHGN architecture need to be implemented. The deployment of mHGN in several risky areas of tornados can then be expected as an alternative way of reducing damages, losses, and costs.

Keywords: Graph Neuron, Hierarchical Graph Neuron, Multidimensional Hierarchical Graph Neuron, Natural Disaster Forecast, Tornado Forecast

1   Introduction

The latest storms that hit several countries across the gulf part of American continent have given a clear evidence that natural disasters are real, and they will occur regularly. Although the upcoming of a storm can be monitored and people are usually notified many hours before it arrives, damages, injures, and casualties are usually still involved. The reason to this is maybe because not only the upcoming is important, but also the lack of preparation of people contributes to those damages and casualties. The clear evidence to this is the two tsunamis: in Indonesia in 2004, and in Japan in 2011. The situation is worse when the disaster is a tornado, because it occurs without an adequate notification (siren, alarm, or radio announcement). In the US [1] [2] the number of occurrences of a tornado is the highest compared to those in other countries.

The reason why the occurrence of natural disasters cannot be forecasted is due to its randomness. Furthermore, many people are not aware of the precondition of its occurrence, and they are not prepared to face that. Such a randomness is not the case of a tornado. In the US, tornados are not random. Most of tornados occurred between April and June. The area of occurrences is in the southeast area of the US. Nevertheless, every single tornado appears very suddenly. Prior to its appearance there is no sign that can be recognized beforehand, for instance at least 10 hours earlier, so that people have adequate time to get away of it. As the result, damages, losses, and costs totally cannot be predicted.

Researchers have worked on handling natural disasters. The general term that they have developed is called disaster management. The management deals with prior, while, and post disaster. The most difficult part dealing with natural disaster, such as a tornado, is to forecast it. It is true that some kind of early warning system has been built in many countries. In the US, a siren is started when the system detects an upcoming of a tornado. However, the time distance between the siren and the occurrence of tornado is very short. It is usually less than 30 minutes. Such period of time is inadequate for people to protect themselves from the fierce of tornado's force. For centuries researchers have worked on discovering ways to forecast the upcoming of a natural disaster. Some of them are still at the stage of now-casting [3] [4] [5] [6] [7], not yet forecasting. Some believe that the most difficult part of forecasting natural disasters is producing the mathematical formulas of it [8].

So, it is a strong indication that other methods than mathematical formulas to forecast a tornado are required. A number of researchers [8] have developed an artificial intelligent technology to forecast an upcoming tornado. Although there is no definite mathematical functions that can be used to determine the condition of a tornado, wind-speed, wind-direction, air-temperature, and air-pressure that constitute a tornado, are all caused by physical states [9]. It means that the time-series of several physical values of wind-speed, wind-direction, air-temperature, and air-pressure determine particular tornado condition. It can then be figured out that the upcoming of a tornado is generally produced by particular physical patterns.

The physical patterns that have been used in mHGN [8] were not yet ready for forecasting tornados, although mHGN is able to recognize incomplete alphabetical patterns [9]. There are two major issues why mHGN is not ready to forecast tornados. First, the physical values of weather data need to be represented using sophisticated scheme so, that false positive and true negative rates can be reduced significantly. Second, the occurrence of a tornado not at the location where the measuring sensors of weather for wind-speed, wind-direction, air-temperature, and air-pressure are located. Therefore, a sophisticated approached is required so that the patterns from previous occurrences of tornados can be used as the training pattern for future purposes.

In this paper, those two issues are elaborated in order to improve the accuracy of mHGN in forecasting tornados.

2   Tornado Recognition

Natural disasters have become common issues in the world. United Nation's UNISDR has been the primary organization that works tightly with disasters around the world. Since natural disasters have occurred in many countries, people across the world are hand in hand helping each other in facing natural disasters. This situation has helped researchers in gaining data from sensors scattered in the world. Researchers have started since long time ago investigating new approaches in forecasting natural disasters. As many parameters and conditions need to be considered, many issues related to forecasting natural disasters need to be investigated further.

The possibility of the occurrence of a natural disaster varies, but it is still there in most parts of the world. For instance, the tsunami that hit Indonesia in 2004 had never been experienced by Indonesians for hundreds of years. This situation applies for other natural disasters. Some countries have experienced natural disasters more than others [8]. This situation has been the main reason why a country such as the US has spent a lot of efforts to deal with natural disasters. When the circumstances of a tornado to come up in an area have reached, it is very likely that the area will suffer from a tornado. Most tornados occurred in the US have caused damages, losses, and costs. However, this does not mean that only the US must concern with the occurrence of tornados.

The randomness of the occurrence of a natural disaster is not only in terms of the location, but also in terms of the time and the severity. However, the location and the time (season) of tornados to occur is generally the same. Previous tornados occur between April and June, and the most countries that have been hit are those in the southeast area of the US. Although many researchers in opinion that the severity and the average magnitude of natural disasters have increased since the last decade. However, it is still not clear how severe future natural disasters might be. The impossibility of measuring, or predicting the severity of natural disasters, has been the major cause of the difficulties in anticipating their occurrences. Many other researchers have suggested that, one way to deal with the randomness of the occurrence of natural disasters is through a disaster forecaster.

Early warning system [2] and now-casting [1] [10] [11] [5] [3] [6] that have been investigated and developed are not yet able to help people avoiding and mitigating natural disaster. If the time frame of detecting a tornado is short, people will not be able to save themselves away from the tornado. For instance, the forecasting approach that they [12] [7] [4] have attempted is able to forecast the disaster, but only within one hour time frame. Despite those efforts of researchers, Sorensen [2] argues that in terms of prediction and forecasting, still no radical breakthroughs have occurred in the past twenty years. Due to its complexities, most natural disaster researchers are working on technologies that are not focusing on the forecasting techniques. Rather, they are concerned with how natural disaster alerts can be disseminated to the public [2]. They suggest that most difficult part in facing a natural disaster is about how to handle people when a natural disaster occurs. Additionally, most common recommendation for an early warning system is "how to evacuate." In relation with natural disaster alerts, it is also important that special attention must be taken for those who have disabilities.

It seems to be that researchers have tried to find an appropriate approach for working on three areas: natural disaster forecaster, now-casters, or early warning systems. However, they [2] [13] [10] also still integrate their system with disaster management systems. Even Doong et al. [13] suggest that the success of a disaster mitigation concept lies in the quality of the disaster management. This shows that their approach alone is not yet adequate to handle natural disasters nor tornados. The potential reason to this case is the fact that a system for handling tornados requires very complicated mathematical analysis. Many parameters and values need to be considered and included in their calculation [4] [5] [1], and it is time consuming [11], but the system must run fast [1], that it can be used to warn people as early as possible.

As already mentioned, although the randomness of the occurrences of a natural disaster has caused difficulties in handling it, the development of every natural disaster still follows natural science characteristics and rules. Each part of a natural disaster of tornado owns specific location, time, patterns and characters. A tornado develops its twist due to hot and cold winds that move from the opposite directions. Not only the opposite winds play a role in developing a tornado, specific air pressure and air temperature are also significant contributors for a tornado's development. Yet, the difficulty to gain the measured values of those characteristics has become a new challenge in recognizing tornados before it turns up.

The steps that a tornado builds before its strong and winding wind can be regarded as a pattern. It means that the recorded data from previous tornado disasters plays a big role in recognizing it. Therefore, the data must be kept properly. The data is the important source of clues for researchers to analyze the pattern of a tornado. When patterns of tornados can be memorized, it is a strong possibility that when one of the patterns turns up, a system that can recognize patterns can be used to recognize a tornado early before it becomes a strong and destructive one. Such patterns are the most important part of mHGN for forecasting tornados hours before they strike. Unfortunately, the data provided by weather stations such as data of NOAA is not so accurate. The location of weather sensors is not exactly where the previous tornados occur. Before discussing the approach through the deployment of mHGN, it is still important to review the fundamental concept of mHGN.

3   Multidimensional Hierarchical Graph Neuron (mHGN)

The purpose of describing mHGN [9] in this paper again, is to provide an easy way for readers to review its concept and its principle.

The need for solving multidimensional problems has been discussed since a long time ago. People are aware of that to handle complex problems, values taken from numerous dimensions must be considered and calculated. Otherwise, the result that comes up after the calculation—analyzing just a few parameters—cannot be considered correct. In most cases, such a condition will produce very high false positive and true negative error rates. Another issue related to solving multidimensional problems is the

method that will be implemented. In a complex system, not only the number of dimensions is large, but how all the dimensions are interrelated to each other, or independent on one another, is often not clear.

Natural disaster system is a suitable example as a multidimensional system. Therefore, forecasting natural disasters can also considered as solving a multidimensional problem. Not only the longitude and the latitude determine the condition of natural disasters, air-temperature, air-pressure, air-humidity, wind-direction, and wind-speed also play a big role in causing natural disasters of tornados. A problem that still exists is the interdependency amongst those tangible and intangible values (industrial development, people movement, etc.). It is so difficult to figure out a formula that constitutes such interdependency. This is, therefore, a strong indication that such multidimensional problems may be solved using artificial intelligent approaches such as mHGN.

3.1   Experiment Results

For the experiment, each neuron (GN) is operated by a thread. Various 2D-, 3D-, 4D- and 5D-pattern recognition have been scrutinized. The compositions used in the experiment are: 15X15 mHGN, 5X15X15 mHGN, 5X5X15X15 mHGN, and 5X5X5X15X15 mHGN respectively. For instance, in the 15X15 pattern recognition the composition requires: $225 + 195 + 165 + 135 + + 105 + 75 + 45 + 15 + 13 + 11 + 9 + 7 + 5 + 3 + 1 = 1009$ neurons per value of data. As for creating patterns, binary data is used, then two values (i.e. 0 and 1) of data are required. Therefore, 2018 neurons are deployed in the 15X15 mHGN composition. So, 2018 threads have been run in parallel during this 2D pattern recognition. By using threads, the activity of neurons is simulated so that the functionalities are close to the real neuron functionalities.

The experiment has worked on all the patterns of 26 alphabetical figures. Following the composition of the neurons, the alphabet patterns consist of 15X15 pixels. For the training purpose, the mHGN is first fed one-cycle with all the 26 non-distorted patterns. The order of the patterns during the training phase has been determined randomly. Then, to acquire the recognition results the mHGN is fed with a lot of randomly distorted patterns of alphabets. The recognizing accuracy is taken by calculating the average value of the results.

For the sake of the experiment, 20 distorted patterns for each alphabetical figure have been prepared. After acquiring the results, the experiment is repeated 10 times with the same steps, but each time the mHGN is trained with 26 patterns of alphabetical figures with randomly different order. So, for each alphabetical figure for particular percentage of distortion, in total 200 distorted patterns have been prepared as testing patterns.

There are 7 levels of distortion that have been tested, they are: 1.3%, 2.7%, 4.4%, 6.7%, 8.0%, 8.9%, and 10.7%. These levels have been so chosen based on the number of distorted pixels. The sizes of pixels represent the factor and the non-factor of the dimension of the patterns. By doing so, we can observe all the possibilities of distortion. So, in total there are 5200 (26 x 20 x 10) randomly distorted testing patterns. The following Figure 1 shows 5 samples of different orders of the training patterns:

1	E	N	W	L	I	S	P	G	H	J	D	Y	A	X	Q	R	C	M	F	V	O	T	U	K	Z	B
2	R	P	J	S	O	Q	D	V	C	K	L	E	F	G	X	Y	A	T	Z	B	U	W	T	H	M	N
3	G	B	H	R	Z	C	I	Y	X	S	J	K	D	A	N	T	Q	V	E	W	F	U	P	O	L	M
4	L	N	I	F	R	X	B	K	O	C	T	Z	A	Y	G	V	U	H	P	J	Q	S	W	E	D	M
5	C	E	T	U	N	R	H	Y	G	D	B	K	F	M	I	X	V	S	Q	J	Z	W	O	A	L	P

Fig. 1. Five different randomly ordered alphabets.

The following shows some results taken from testing 4.4% randomly distorted patterns, and the mHGN was previously stored with alphabetical figure patterns, and the order was IEFXMQYJHPDKTORZCUALBGVWNS. The value on the right side of each alphabet show the portion (percentage) of the pattern that is recognizable as the corresponding alphabet.

PATTERNS RANDOMLY DISTORTED 4.4 %

Patterns Stored	Distorted Pattern	Recognised patterns and their recognized portion (%) from 20 different randomly distorted patterns																				Recognised Correctly
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
I	A	A 9	A 8	A 8	A 54	A 13	A 3	A 8	A 13	A 16	A 6	A 15	A 14	A 15	A 14	A 14	A 2	A 1	A 54	A 17	A 15	20
E	B	B 3	B 4	B 36	B 37	B 11	B 36	B 11	B 11	B 37	B 10	B 5	B 36	B 10	B 12	B 11	B 6	B 9	B 11	B 36		20
F	C	C 40	C 7	C 38	C 15	C 8	C 39	C 8	C 7	C 8	C 15	C 7	C 0	C 14	C 15	C 40	C 14	C 1	C 14	C 39		20
X	D	D 26	D 10	D 4	D 27	D 9	D 10	D 10	D 2	D 26	D 12	D 25	D 11	D 5	D 12	D 6	D 9	D 9	D 10	D 11	D 10	20
M	E	E 10	E 6	E 30	E 29	E 12	E 11	E 11	E 10	E 10	E 29	E 11	E 10	E 10	E 11	E 12	E 29	E 31	E 6	E 10	E 9	20
Q	F	F 20	F 4	F 19	F 3	F 20	F 9	F 7	F 8	F 4	F 1	F 7	F 21	F 19	F 22	F 8	F 9	F 21	F 8	F 4	F 8	19
Y	G	G 51	G 14	G 51	G 13	G 15	G 51	G 14	G 50	G 7	G 51	G 14	G 15	G 6	G 15	G 7	G 3	G 14	G 6	G 7		20
J	H	H 8	H 14	H 54	H 14	H 15	H 56	H 15	H 14	H 7	H 54	H 13	H 3	H 16	H 55	H 14	H 54	H 9	H 15	H 55	H 54	20
H	I	I 54	I 6	I 1	I 53	I 6	I 14	I 14	I 55	I 15	I 15	I 54	I 14	I 15	I 54	I 6	I 15	I 14	I 16	I 5		20
P	J	J 54	J 5	J 5	J 14	J 15	J 13	J 54	J 14	J 55	J 6	J 55	J 15	J 15	J 6	J 54	J 55	J 14	J 13	J 14	J 3	20
D	K	K 4	K 7	K 8	K 8	K 6	K 6	K 8	K 5	K 8	H 22	K 9	K 7	K 6	K 4	K 2	H 22	K 9	K 7	K 5		18
K	L	L 9	L 11	L 5	L 38	L 11	L 11	L 11	L 39	L 6	L 40	L 12	L 39	L 10	L 6	L 5	L 10	L 7	L 10	L 12	L 38	20
Order Type 0 — T	M	M 22	M 19	M 21	M 4	M 21	M 22	M 8	M 4	M 8	M 6	M 7	M 6	M 5	M 7	M 5	M 4	M 7	M 7	M 6	M 7	20
O	N	N 19	N 4	N 3	H 7	N 19	N 8	N 8	N 4	N 7	N 6	N 7	N 8	H 18	N 8	N 6	N 7	N 8	N 5	N 7	N 5	18
R	O	O 54	O 16	O 15	O 14	O 14	O 7	O 56	O 14	O 54	O 8	O 3	O 16	O 14	O 16	O 55	O 54	O 6	O 15	O 7	O 54	20
Z	P	P 10	P 32	P 10	P 9	P 9	P 10	P 5	P 6	P 5	P 33	P 10	P 0	P 8	P 10	P 10	P 32	P 5	P 9	P 11	P 11	20
C	Q	Q 44	Q 15	Q 4	Q 15	Q 45	Q 15	Q 16	Q 13	Q 14	Q 15	Q 15	Q 15	Q 45	Q 6	Q 44	Q 14	Q 45				20
U	R	R 11	R 12	R 12	R 5	R 12	R 11	R 38	R 5	R 6	R 10	R 10	R 7	R 37	R 10	R 37	R 37	R 38	R 11	R 2	R 12	20
A	S	S 14	S 15	S 8	S 14	S 13	S 3	S 7	S 13	S 14	S 46	S 45	S 13	S 15	S 46	S 7	S 7	S 13	S 14	S 15		20
L	T	T 14	T 15	T 14	T 15	T 15	T 16	T 13	T 7	T 15	T 55	T 15	T 14	T 15	T 55	T 16	T 54	T 55	T 54	T 14	T 14	20
B	U	U 4	U 7	U 10	U 7	U 10	U 11	U 11	U 9	U 11	U 5	U 30	U 11	U 5	U 30	U 4	U 10	U 10	U 11	U 2		20
G	V	V 56	V 14	V 54	V 15	V 54	V 8	V 55	V 55	V 6	V 14	V 14	V 9	V 2	V 15	V 1	V 56	V 16	V 14	V 16	V 15	20
V	W	W 6	W 14	W 30	W 29	W 13	W 13	W 13	W 14	W 6	W 12	W 14	W 11	W 13	W 13	W 7	W 30	W 30	W 11	W 12		20
W	X	X 15	X 53	X 16	X 15	X 15	X 14	X 15	X 14	X 14	X 14	X 7	X 6	X 16	X 54	X 15	X 14	X 54	X 14	X 13	X 54	20
N	Y	Y 15	Y 16	Y 15	Y 13	Y 14	Y 51	Y 8	Y 2	Y 51	Y 13	Y 15	Y 14	Y 13	Y 52	Y 16	Y 15	Y 15	Y 5	Y 15		20
S	Z	Z 7	Z 15	Z 15	Z 14	Z 13	Z 14	Z 16	Z 17	Z 0	Z 51	Z 14	Z 51	Z 50	Z 7	Z 6	Z 7	Z 51	Z 14	Z 13	Z 51	20

Fig. 2. The result of all the 26 alphabetical patterns that are twenty times 4.4% randomly distorted.

The following shows 10 samples of distorted patterns of the alphabetical figure of "A" taken from the experiment of recognizing 5.8% randomly distorted patterns.

After collecting the results taken from testing 5200 patterns we can summarize how accurate the mHGN is, in recognizing different levels of distortion of 26 alphabets. The summary is taken based on the average accuracy values from all the steps. The following shows the summarized result taken from testing distorted patterns using five-dimensional 5X5X5X15X15 mHGN.

It can be seen from Figure 4 in the last column that the mHGN is able to recognize 91% of the 10.7% distorted patterns of 26 alphabetical figures. Some alphabetical figures of A, C, E, G, I, J, L, O, S, T, U, V, X, Y, Z, are even 100% recognizable. Other patterns of alphabetical figures of H, K, M, N, are not very well recognized because they are visually and physically very similar. In fact, if this architecture is used to recognize different states of the same alphabet, such as regular-A, bold-A, and italic-A as the same alphabet, then mHGN will be able to gain better accuracy values.
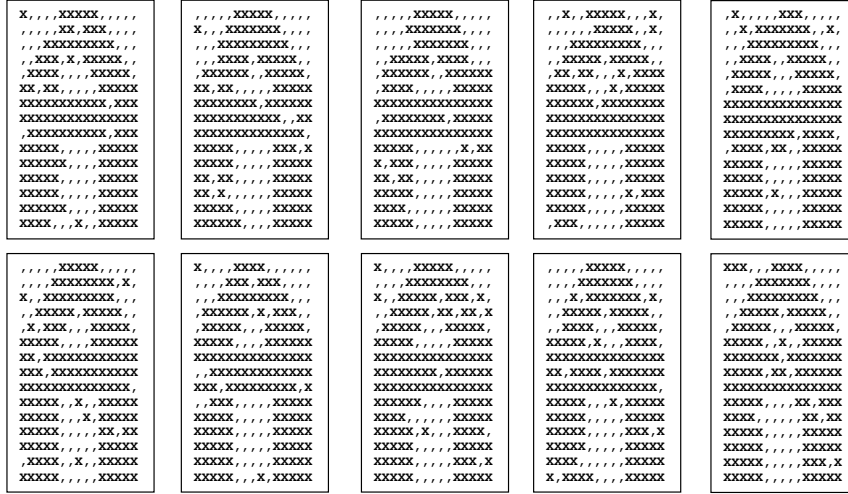
```
x,,,,XXXXX,,,,,   ,,,,,XXXXX,,,,,   ,,,,,XXXXX,,,,,   ,,X,,XXXXX,,,X,   ,X,,,,,XXX,,,,,
,,,,,XX,XXX,,,,   x,,,XXXXXXX,,,,   ,,,,,XXXXXXX,,,   ,,,,,XXXXX,,X,,   ,,X,XXXXXXX,,X,
,,,XXXXXXXXX,,,   ,,,XXXXXXXXX,,,   ,,,,,XXXXXXX,,,   ,,,XXXXXXXXX,,,   ,,,XXXXXXXXX,,,
,,XXX,X,XXXXX,,   ,,,XXXX,XXXXX,,   ,,XXXXX,XXXXX,,   ,,XX,XX,,,X,XXXX   ,,XXXX,,XXXXX,,
,XXXX,,,,XXXXX,   ,XXXXXX,,XXXXX,   ,XXXX,,,,,XXXXX   XXXXX,,,X,XXXXX   ,XXXXX,,,XXXXX,
XX,XX,,,,,XXXXX   XX,XX,,,,,XXXXX   ,XXXX,,,,,XXXXX   XXXXXX,XXXXXXXX   ,XXXX,,,,,XXXXX
XXXXXXXXXX,XXX    XXXXXXXXX,XXXXXX   XXXXXXXXXX,XXXX   XXXXXXXXXXXXXXX   XXXXXXXXXXXXXXX
XXXXXXXXXXX,XX    XXXXXXXXXX,,XX    XXXXXXXXX,XXXX    XXXXXXXXXXXXXXX   XXXXXXXXXXXXXXX
,XXXXXXXXX,XXX    XXXXXXXXXXXXX,    XXXXXXXXXXXXXX    XXXXX,,,,,XXXXX   XXXXXXXX,XXXX,
XXXXX,,,,,XXXXX   XXXXX,,,,,XXX,X   XXXXX,,,,,,X,XX   ,XXXX,XX,,XXXXX   ,XXXX,XX,,XXXXX
XXXXXX,,,,XXXXX   XXXXX,,,,,XXXXX   X,XXX,,,,,XXXXX   XXXXX,,,,,XXXXX   XXXXX,,,,,XXXXX
XXXXX,,,,,XXXXX   XX,XX,,,,,XXXXX   XX,XX,,,,,XXXXX   XXXXX,,,,,XXXXX   XXXXX,,,,,XXXXX
XXXXX,,,,,XXXXX   XX,X,,,,,,XXXXX   XXXX,,,,,,XXXXX   XXXXX,,,,X,XXX    XXXXX,X,,,XXXXX
XXXXXX,,,,XXXXX   XXXXX,,,,,XXXXX   XXXX,,,,,,XXXXX   XXXXX,,,,,XXXXX   XXXXX,,,,,XXXXX
XXXX,,,X,,XXXXX   XXXXX,,,,XXXXX    XXXXX,,,,,XXXXX   ,XXX,,,,,,XXXXX   XXXXX,,,,,XXXXX
```

```
,,,,,XXXXX,,,,,   x,,,,XXXX,,,,,,   x,,,,XXXXX,,,,,   ,,,,,XXXXX,,,,,   XXX,,,XXXX,,,,,
,,,XXXXXXXX,X,    ,,,,XXX,XXX,,,,   ,,,,,XXXXXXX,,,   ,,,,,XXXXXXX,,,   ,,,,,XXXXXXX,,,
X,,XXXXXXXXX,,,   ,,,XXXXXXXXX,,,   X,,XXXXX,XXX,X,   ,,,X,XXXXXXX,X,   ,,,XXXXXXXXX,,,
,,XXXXX,XXXX,,    ,XXXXX,X,XXX,,    ,,XXXXX,XX,XX,X   ,,XXXXX,XXXXX,,   ,,XXXXX,XXXXX,,
,X,XXX,,,XXXXX,   ,XXXXX,,,XXXXX,   ,XXXXX,,,XXXXX,   ,,XXXX,,,XXXXX,   ,XXXXX,,,XXXXX,
XXXXX,,,,,XXXXXX  XXXXX,,,,,XXXXXX  XXXXX,,,,,XXXXX   XXXXX,X,,,XXXX,   XXXXX,,X,,XXXXX
XX,XXXXXXXXXXXX   ,,XXXXXXXXXXXX    XXXXXXXXXXXXXX    XXXXXXXXXXXXXXX   XXXXXXX,XXXXXXX
XXX,XXXXXXXXXX    XXX,XXXXXXXXX,X   XXXXXXXXX,XXXXX   XX,XXXX,XXXXXXX   XXXXX,XX,XXXXXX
XXXXXXXXXXXXX,    ,,XXX,,,,,XXXXX   XXXXXXXXXXXXXX    XXXXXXXXXXXXXX,   XXXXXXXXXXXXXX
XXXXX,,X,,XXXXX   XXXXX,,,,,XXXXX   XXXXX,,,,,XXXXX   XXXXX,,,X,XXXXX   XXXXX,,,,XX,XXX
XXXXX,,,X,XXXXX   XXXXX,,,,,XXXXX   XXXXX,X,,,XXXX,   XXXXX,,,,,XXX,X   XXXX,,,,,,XX,XX
XXXXX,,,,,XX,XX   XXXXX,,,,,XXXXX   XXXXX,,,,,XXXXX   XXXXX,,,,,XXX,X   XXXXX,,,,,XXXXX
XXXXX,,,,,XXXXX   XXXXX,,,,,XXXXX   XXXXX,X,,,XXXX,   XXXXX,,,,,XXX,X   XXXXX,,,,,XXXXX
,XXXX,,,X,,XXXXX  XXXXX,,,,,XXXXX   XXXXX,,,,XXX,X    XXXX,,,,,,XXXXX   XXXXX,,,,XXX,X
XXXXX,,,,,XXXXX   XXXXX,,,X,XXXXX   XXXXX,,,,,XXXXX   X,XXXX,,,,XXXXX   XXXXX,,,,,XXXXX
```

Fig. 3. Ten different randomly 5.8% distorted patterns of alphabetical figure of "A"

5X5X5X15X15 Patterns		Distortion (%)						
		1.3	2.7	4.4	6.7	8.0	8.9	10.7
Recognition Accuracy for Each Pattern (%)	A	100	100	100	100	100	100	100
	B	100	100	100	100	98	97	94
	C	100	100	100	100	100	96	100
	D	100	100	100	100	100	100	98
	E	100	100	100	100	100	100	100
	F	100	99	94	89	83	85	74
	G	100	100	100	100	100	100	100
	H	100	100	89	67	48	50	55
	I	100	100	100	100	100	100	100
	J	100	100	100	100	100	100	100
	K	100	100	98	81	70	72	67
	L	100	100	100	100	100	100	100
	M	100	100	93	76	55	66	49
	N	100	100	97	77	63	60	55
	O	100	100	100	100	100	100	100
	P	100	99	87	79	80	81	81
	Q	100	100	100	100	100	94	99
	R	100	100	100	95	100	99	95
	S	100	100	100	100	100	100	100
	T	100	100	100	100	100	100	100
	U	100	100	100	100	100	100	100
	V	100	100	100	100	100	100	100
	W	100	100	100	100	99	98	92
	X	100	100	100	100	100	100	100
	Y	100	100	100	100	100	100	100
	Z	100	100	100	100	100	100	100
Average		100	100	98	95	92	92	91

Fig. 4. The summary of the result using 5X5X5X15X15 mHGN [14]

The following figure shows the differences of recognition accuracy amongst 15X15, 5X15X15, 5X5X15X15, and 5X5X5X15X15 mHGN architectures when recognizing 10.7% distorted patterns of alphabets.

Comparison Result		Distortion = 10.7 %			
		15X15	5X15X15	5X5X15X15	5X5X5X15X15
Recognition Accuracy for Each Pattern (%)	A	99	100	100	100
	B	58	69	92	94
	C	67	93	94	100
	D	78	92	94	98
	E	85	80	100	100
	F	61	71	81	74
	G	87	98	100	100
	H	23	63	69	55
	I	95	100	100	100
	J	77	95	100	100
	K	68	59	84	67
	L	50	80	100	100
	M	38	36	35	49
	N	53	42	63	55
	O	100	100	100	100
	P	61	59	75	81
	Q	63	73	73	99
	R	79	90	95	95
	S	78	97	100	100
	T	93	95	100	100
	U	89	84	85	100
	V	100	100	100	100
	W	75	82	98	92
	X	85	100	100	100
	Y	100	100	100	100
	Z	99	100	100	100
Average		75	83	90	91

Fig. 5. Differences of recognition accuracy amongst four different architectures

3.2   Time-Series in Pattern Recognition

Recognizing patterns of time series problem utilizes data that have previously been recorded regularly in timely manner [14]. For instance, if the parameter that needs to be recorded is a single value, and the recording step is every six hours, then there will be 4 values recorded every day. In order to constructs the recorded values as a pattern, the data representation of the recorded values need to be developed so, that they can fit into a pattern recognition architecture. The following figure shows six ways of representing recorded data for 8 values of measurement.

Definition: The distance between two values is the number of different bits between them. For instance, the distance between 001 and 110 is 3, whereas the distance between 00000001 and 10000000 is 2. Some people know it as Hamming Distance.

It can be seen from figure 6 that the data (value) is represented using binary numbers, and there are six possible data representations (Ver1 till Ver6). For the Ver1 (Binary Code Decimal), the distance between adjacent values varies. It is therefore not suitable for mHGN. For the Ver2 (Grey Code), the distance between two adjacent values is constant, which is one. However, the distance between Value 2 and 7 is also one. This is not suitable, as for mHGN the distance of 1 also means that the two values are very

close to each other. In fact, value 2 and 7 are very different and very far from each other. Again, this is not suitable for mHGN. For the Ver3 and Ver4 (Ring Counter), the distance between any two values is constant, which is two. These are again not suitable for mHGN. For the Ver5 and Ver6 (Johnson Counter) the distance between adjacent values is constant, which is one. Additionally, the distance between any two values is linear with the value differences.

Value	Ver1	Ver2	Ver3	Ver4	Ver5	Ver6
1	000	000	00000001	10000000	00000001	10000000
2	001	001	00000010	01000000	00000011	11000000
3	010	011	00000100	00100000	00000111	11100000
4	011	010	00001000	00010000	00001111	11110000
5	100	110	00010000	00001000	00011111	11111000
6	101	111	00100000	00000100	00111111	11111100
7	110	101	01000000	00000010	01111111	11111110
8	111	100	10000000	00000001	11111111	11111111

Fig. 6. Six examples of data representation for 8 levels of value

It seems to be that the Ver5 and Ver6 are the most suitable data representation that can be used with mHGN. However, such data representation will not efficiently utilize the binary combination. Using 3-bit data, only 3/8 or 0.375 is the occupation ratio. Using 4-bit data is the occupation ratio 4/16 or 0.25. The occupation ratio is 5/32 or 0.15625 when using 5-bit data. The facts show that the Ver5 and Ver6 data representation will produce less occupation ratio, the more bits is used. This is an indication that due to such an occupation ratio the pattern recognizer will have less recognition accuracy, the more bits is used. So, these are again not suitable for mHGN. The following is a better data representation.

In figure 7, there are three examples of 3-bit, 4-bit, and 5-bit data representation. It is shown that the distance between any adjacent levels in all samples is constant, which is one. Furthermore, between any two values which have value difference of two, the distance is also constant, which is two. Last, the distance between any two values which have value difference of three, the distance is again constant, which is three. For mHGN, such constant distances of 1, 2, and 3 are adequate to be used in tornado recognition architecture. Another characteristic of these data representations is that the representation is cyclic. It means that, if it is required the order of binary representation can be modified circularly without affecting the characteristic related to the distances. Using such better data representations, in all examples is the occupation ratio constantly 0.75. With such a constant occupation ratio the pattern recognizer will have constant recognition accuracy, regardless how many bits of data representation is used.

The figure 8 shows an example of recorded data taken from a single location measurement and each value has 3X5-bit values.

Value	3-bit	4-bit	5-bit
1	101	0101	00101
2	**100**	**0100**	**00100**
3	110	0110	00110
4	010	1110	01110
5	**011**	**1111**	**01111**
6	001	1101	01101
7		1001	01001
8		**1000**	**01000**
9		1010	01010
10		0010	11010
11		**0011**	**11011**
12		0001	11001
13			11101
14			**11100**
15			11110
16			10110
17			**10111**
18			10101
19			10001
20			**10000**
21			10010
22			00010
23			**00011**
24			00001

Fig. 7. Three examples of a better data representation for 3-, 4-, and 5-bit binary values

```
1   0   1   1   0   0   1   1   0   0
1   1   1   1   0   1   1   1   1   0
0   0   0   1   1   1   1   1   0   1
1   1   1   0   0   1   0   0   1   0
1   0   0   1   0   1   1   0   0   0
1   0   1   0   1   0   1   1   0   0
1   1   1   0   1   1   1   1   1   1
0   1   1   1   0   0   0   1   0   1
1   1   0   0   1   1   1   0   1   1
1   1   0   0   0   0   0   1   0   1
0   1   0   0   0   0   1   0   1   1
0   1   1   1   1   1   1   0   1   1
1   0   1   0   1   1   1   1   1   0
0   1   1   1   1   1   0   0   0   1
0   1   1   0   1   1   0   0   1   1
```

00.00 03.00 06.00 09.00 12.00 15.00 18.00 21.00 00.00 03.00
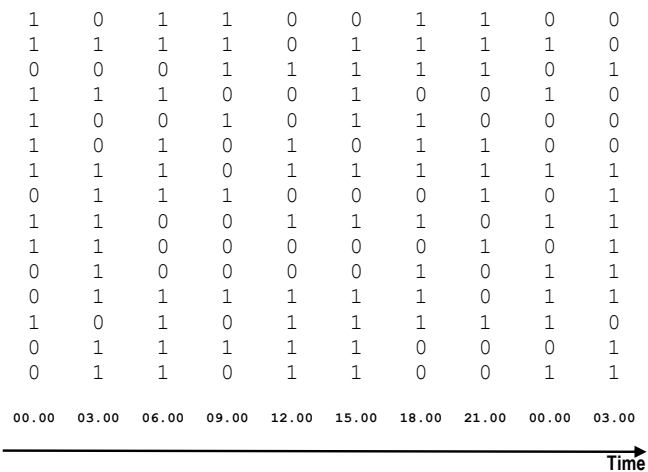
→ **Time**

Fig. 8. Time series data of 15-bit values build a 2D-Pattern

It can be seen from figure 8 that the recorded values from parameter of 15-bit data construct a two-dimensional pattern of 10X15 architecture. Utilizing these recorded data, the pattern recognizer can forecast a tornado 6 hours earlier, when the same tornado occurs again. It means that if values have been recorded and the same pattern is recognized by the pattern recognizer, then the tornado is forecasted to occur again within 6-hour time with around 90% of accuracy.

4 Multidimensional Graph Neuron for Tornado Forecasting

In the previous section, time series value is described and represented so, that it can be forecasted through utilizing a pattern recognition, such as mHGN. In case of tornado forecasting, single parameter in a location, such as air-pressure, is not the only value that determine the occurrence of a tornado in the location within 6-hour time. Several other parameters, such as wind-speed, wind-direction, air-temperature, and air-humidity, play a big role in the occurrences as well. It means that the number of levels or a measured value will increase according to the number of parameters. In case 5 parameters need to be measured and each parameter contains 5-bit value, the required pattern structure would be 10X25.

Also described in the previous section that measuring a parameter at particular point of location for several periods of time will generate a two dimensional pattern. If a series of points of the location need to be measured for several period of time, then the measured values will become a three dimensional pattern. The following figure depicts how some part of it will look like.
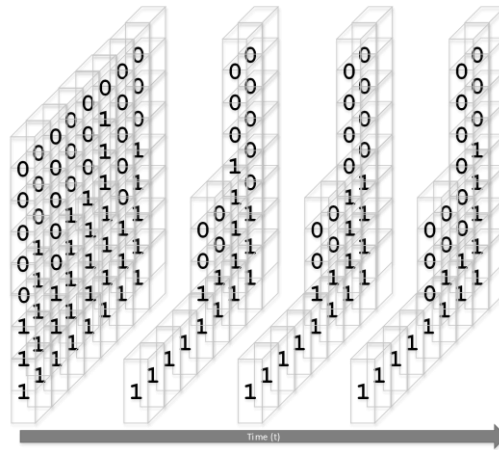
Fig. 9. A row of data of 8-bit value build a 3D-Pattern

Also described in the previous section that measuring parameters at particular point of location for several periods of time will generate a two dimensional pattern. If a series and linear of locations need to be measured for several periods of time, then the measured values will become a three dimensional pattern. If the location that need to

be measured is an 2D area, then the measured values will generate a 4D pattern. Furthermore, if the location that need to be measured is a 3D area, then the measured values will generate a 5D pattern.

4.1 The Architecture of mHGN for Time-Series Tornado Data

The utilization of mHGN has introduced a new approach that a local tornado forecast can be operated using small and cheap components. The values of air-temperature, air-humidity, air-pressure, wind-speed, and wind-direction can be gained through ordinary sensors. The area that is covered by those sensors can be a 3D area, because such small sensors can be easily mounted in valleys or hills, or even vehicles. The sensors can be embedded in a tiny computer, such as Raspberry Pi. The tiny computer will be responsible to run several GNs. The values taken from the sensors will then be worked out within the GNs. The connectivity of neurons is developed within a tiny computer and through the interconnectivity of the tiny computers.

In short, to build a tornado forecast for particular location, five parameters need to be measured. They are: wind-speed, wind-direction, air-temperature, air-humidity, and air-pressure. So, if one parameter is represented through 5-bit binary data, then for the measurement of 5 parameters 25-bit data is needed. For the time series, 15 series of measurement will be carried out. For an area that needs to be protected by mHGN, 3X3X3 measurement points will be deployed. So, the mHGN dimension will be 3X3X3X25X15.

The positions of the 3X3X3 GNs will form a cylinder shape. In the cylinder, there will be three layers of circles. Each layer contains 9 GNs, in which 8 GNs will be on the border of the circle, and one GN will be located in the centre of the circle. The following figure shows the architecture of the positions of the sensors.
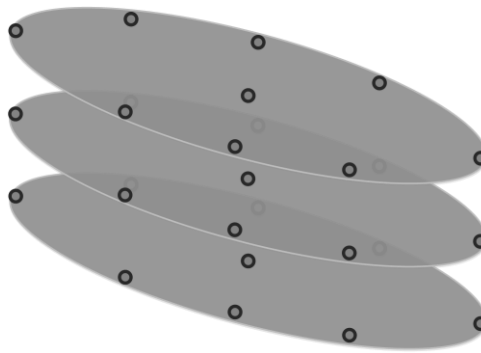
Fig. 10. The architecture of 3X3X3 sensors

The cylinder shape of the architecture has been chosen so, that mHGN still has an ability to recognize the same tornado pattern but developed with the direction different from the ones already stored. For the purpose of training, patterns from the previous tornados will be stored in the mHGN. Each pattern of a tornado will then be stored in

mHGN eight times, following the number of eight major compass directions. It will look like as if the mHGN has stored 8 patterns of tornados. By having eight patterns for each tornado stored in mHGN, whenever the same characteristics of a tornado turn up but from different direction from the already stored ones, mHGN will be able to recognize it.

4.2 Data Handlings for Real Tornados

Two deadliest tornados occurred quite recently are the tornado that struck Joplin, Missouri on May 22, 2011 and the one in Hackleburg–Phil Campbell, Alabama on April 27, 2011. To store the circumstances, several parameters in these areas need to be stored in mHGN. Fortunately, the National Oceanic and Atmospheric Administration (NOAA) provides lots of data of: air-temperature, air-humidity, air-pressure, wind-speed, wind-direction in most areas of the US. These data will be the major source for mHGN to store previous occurrences of tornados. In the case of Joplin, the following are several locations of stations that have recorded those data from their sensors including the map in the state of Missouri.

The following is the list of tornados scale F5/EF5 (the strongest) occurred In the US. The indicator F stands for Fujita and EF stands for extended Fujita. The scale has been named the same as the meteorologist Ted Fujita, who developed the scale.

1. May 4, 2007, Greensburg, Kansas
2. May 25, 2008, Parkersburg–New Hartford, Iowa
3. April 27, 2011, Philadelphia–Preston, Mississippi
4. April 27, 2011, Smithville, Mississippi
5. April 27 2011, Hackleburg–Phil Campbell, Alabama
6. April 27 2011, Tuscaloosa–Birmingham, Alabama
7. April 27, 2011, Rainsville–Sylvania, Alabama
8. May 22 2011, Joplin, Missouri
9. May 24, 2011, El Reno–Piedmont, Oklahoma
10. May 24, 2011, Chickasha–Blanchard–Newcastle, Oklahoma
11. May 24, 2011, Washington–Goldsby, Oklahoma
12. May 20, 2013, Moore, Oklahoma
13. May 31, 2013, El Reno, Oklahoma
14. April 27, 2014, Vilonia, Arkansas

It can be seen from the map above, that based on the close location and the same time frame the data of Joplin's tornado can be used to test the Oklahoma's tornados (five tornados). Similarly, the data of Philadelphia-Preston's tornado or Smithville's tornado can be used to test Alabama's tornados (three tornados).

50

Fig. 11. The locations of F5/EF5 tornados
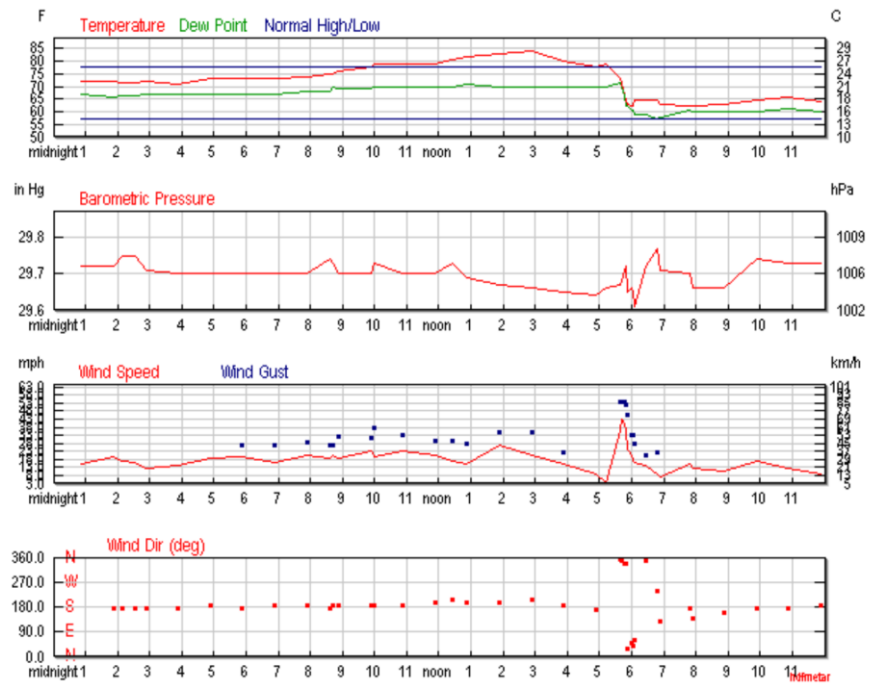
Daily Weather History Graph

Fig. 12. The Weather data of Joplin Station on May 22, 2011

Hourly Weather History & Observations

Time (CDT)	Temp.	Heat Index	Dew Point	Humidity	Pressure	Visibility	Wind Dir	Wind Speed	Gust Speed	Precip	Events	Conditions
12:53 AM	22.2 °C	-	19.4 °C	84%	1006.4 hPa	12.9 km	South	24.1 km/h / 6.7 m/s	-	N/A		Clear
1:53 AM	22.2 °C	-	18.9 °C	81%	1006.4 hPa	12.9 km	South	31.5 km/h / 8.7 m/s	42.6 km/h / 11.8 m/s	N/A		Partly Cloudy
2:10 AM	22.0 °C	-	19.0 °C	83%	1007.3 hPa	12.9 km	South	27.8 km/h / 7.7 m/s	-	N/A		Mostly Cloudy
2:33 AM	22.0 °C	-	19.0 °C	83%	1007.3 hPa	12.9 km	South	25.9 km/h / 7.2 m/s	-	N/A		Scattered Clouds
2:53 AM	22.2 °C	-	19.4 °C	84%	1005.9 hPa	11.3 km	South	20.4 km/h / 5.7 m/s	37.0 km/h / 10.3 m/s	N/A		Clear
3:53 AM	21.7 °C	-	19.4 °C	87%	1005.8 hPa	11.3 km	South	22.2 km/h / 6.2 m/s	-	N/A		Partly Cloudy
4:53 AM	22.8 °C	-	19.4 °C	81%	1005.7 hPa	11.3 km	South	29.6 km/h / 8.2 m/s	40.7 km/h / 11.3 m/s	N/A		Overcast
5:53 AM	22.8 °C	-	19.4 °C	81%	1005.5 hPa	11.3 km	South	31.5 km/h / 8.7 m/s	42.6 km/h / 11.8 m/s	N/A		Overcast
6:53 AM	22.8 °C	-	19.4 °C	81%	1005.6 hPa	9.7 km	South	25.9 km/h / 7.2 m/s	42.6 km/h / 11.8 m/s	N/A		Haze
7:53 AM	23.3 °C	-	20.0 °C	82%	1005.8 hPa	9.7 km	South	33.3 km/h / 9.3 m/s	46.3 km/h / 12.9 m/s	N/A		Haze
8:36 AM	24.0 °C	-	20.0 °C	78%	1007.0 hPa	9.7 km	South	29.6 km/h / 8.2 m/s	42.6 km/h / 11.8 m/s	N/A		Haze
8:43 AM	24.0 °C	-	21.0 °C	83%	1006.7 hPa	9.7 km	South	33.3 km/h / 9.3 m/s	42.6 km/h / 11.8 m/s	N/A		Haze

Fig. 13. The Detailed Weather data of Joplin Station on May 22, 2011
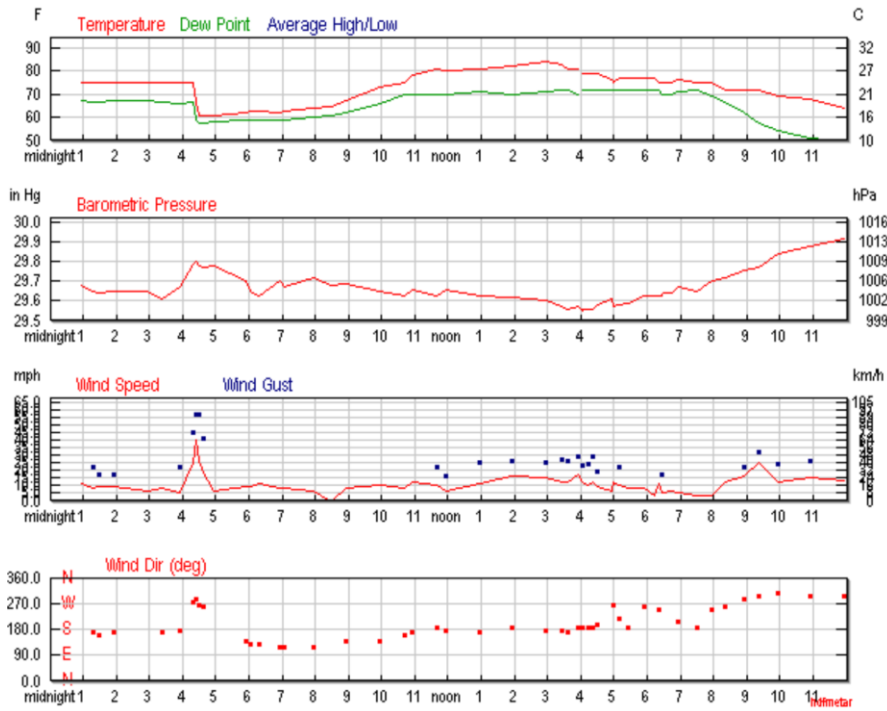

Fig. 14. The Weather data of Philadelphia-Preston Station on April 27, 2011

In order to collect the suitable data that fit with the architecture of mHGN, the locations of the chosen weather stations that the data will be taken from, must build a figure like a circle, and the middle weather station must be located in the area in which a tornado has hit. The following is an excerpt of the data taken from a weather station Joplin in the State of Missouri.

5   Discussion

As is the case with pattern recognition of alphabets, patterns are more or less different to one another. However, in time series measurement data patterns, which are constructed from the measured values of the sensors, can be very similar to one another. Therefore, data representation of measured values before data is fed to the architecture of mHGN plays a big role in having very accurate results. False positive and true negative rate will also be indications to determine the quality of mHGN in forecasting natural disastesr.

The data that will be used to validate this work will be the data taken from different cities and different countries. As mHGN is trained one-cycle only, it is a challenge to choose which data is the right data for the training purpose, or the data is the consolidated data from a number of occurrences. When the appropriate training data has been applied, mHGN will then have a capability to forecast a tornado.

The main thing that needs to be developed for supporting the tornado forecasting technology is a sequence of binary values. Since the analog values taken from sensors (temperature, humidity, pressure, wind speed, and wind direction) need to be precise, so at least 5-bit binary values are required.

The requirements for the sequence of the 5-bit binary values are:

1. The Hamming Distance of any binary value to its first adjacent neighbour value is 1.
2. The Hamming Distance of any binary value to its second adjacent neighbour value is 2.
3. The Hamming Distance of any binary value to its third adjacent neighbour value is 3.
4. The sequence should be cyclic, which means that any binary-value can be either the first or the last member of the sequence.

The following are those mentioned binary values (Fig. 15).

It can be seen from the table above that all four requirements are fulfilled. To prove that the sequence is cyclic, it can be done through inserting the value at the index of 1 underneath the value at the index of 24. It can then be seen that all three requirements of the sequence (about Hamming Distances) are still fulfilled. So, after the index of 24, the sequence can be continued with the values of the index of 1, and 2, and 3, and so forth.

Index	Value	Index	Value
1	00101	**13**	11101
2	**00100**	**14**	**11100**
3	00110	**15**	11110
4	01110	**16**	10110
5	**01111**	**17**	**10111**
6	01101	**18**	10101
7	01001	**19**	10001
8	**01000**	**20**	**10000**
9	01010	**21**	10010
10	11010	**22**	00010
11	**11011**	**23**	**00011**
12	11001	**24**	00001

Fig. 15. The binary values for 5-bit data

	00101	00100	00110	01110	01111	01101	01001	01000	01010	11010	11011	11001	11101	11100	11110	10110	10111	10101	10001	10000	10010	00010	00011	00001
00101	0	**1**	2	3	**2**	1	2	**3**	4	5	**4**	3	2	**3**	4	3	**2**	1	2	**3**	4	3	**2**	1
00100	1	0	1	2	**3**	2	3	**2**	3	4	**5**	4	3	**2**	3	2	**3**	2	3	**2**	3	2	**3**	2
00110	2	**1**	0	1	**2**	3	4	**3**	2	3	**4**	5	4	**3**	2	1	**2**	3	4	**3**	2	1	**2**	3
01110	3	**2**	1	0	**1**	2	3	**2**	1	2	**3**	4	3	**2**	1	2	**3**	4	5	**4**	3	2	**3**	4
01111	**2**	**3**	**2**	**1**	0	**1**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**4**	**5**	**4**	**3**	**2**	**3**
01101	1	**2**	3	2	**1**	0	1	**2**	3	4	**3**	2	1	**2**	3	4	**3**	2	3	**4**	5	4	**3**	2
01001	2	**3**	4	3	**2**	1	0	**1**	2	3	**2**	1	2	**3**	4	5	**4**	3	2	**3**	4	3	**2**	1
01000	**3**	**2**	**3**	**2**	**3**	**2**	**1**	**0**	**1**	**2**	**3**	**2**	**3**	**2**	**3**	**4**	**5**	**4**	**3**	**2**	**3**	**2**	**3**	**2**
01010	4	**3**	2	1	**2**	3	2	**1**	0	1	**2**	3	4	**3**	2	3	**4**	5	4	**3**	2	1	**2**	3
11010	5	**4**	3	2	**3**	4	3	**2**	1	0	**1**	2	3	**2**	1	2	**3**	4	3	**2**	1	2	**3**	4
11011	**4**	**5**	**4**	**3**	**2**	**3**	**2**	**3**	**2**	**1**	**0**	**1**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**
11001	3	**4**	5	4	**3**	2	1	**2**	3	2	**1**	0	1	**2**	3	4	**3**	2	1	**2**	3	4	**3**	2
11101	2	**3**	4	3	**2**	1	2	**3**	4	3	**2**	1	0	**1**	2	3	**2**	1	2	**3**	4	5	**4**	3
11100	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**1**	**0**	**1**	**2**	**3**	**2**	**3**	**2**	**3**	**4**	**5**	**4**
11110	4	**3**	2	1	**2**	3	4	**3**	2	1	**2**	3	2	**1**	0	1	**2**	3	4	**3**	2	3	**4**	5
10110	3	**2**	1	2	**3**	4	5	**4**	3	2	**3**	4	3	**2**	1	0	**1**	2	3	**2**	1	2	**3**	4
10111	**2**	**3**	**2**	**3**	**2**	**3**	**4**	**5**	**4**	**3**	**2**	**3**	**2**	**3**	**2**	**1**	**0**	**1**	**2**	**3**	**2**	**3**	**2**	**3**
10101	1	**2**	3	4	**3**	2	3	**4**	5	4	**3**	2	1	**2**	3	2	**1**	0	1	**2**	3	4	**3**	2
10001	2	**3**	4	5	**4**	3	2	**3**	4	3	**2**	1	2	**3**	4	3	**2**	1	0	**1**	2	3	**2**	1
10000	**3**	**2**	**3**	**4**	**5**	**4**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**1**	**0**	**1**	**2**	**3**	**2**
10010	4	**3**	2	3	**4**	5	4	**3**	2	1	**2**	3	4	**3**	2	1	**2**	3	2	**1**	0	1	**2**	3
00010	3	**2**	1	2	**3**	4	3	**2**	1	2	**3**	4	5	**4**	3	2	**3**	4	3	**2**	1	0	**1**	2
00011	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**2**	**3**	**4**	**5**	**4**	**3**	**2**	**3**	**2**	**3**	**2**	**1**	**0**	**1**
00001	1	**2**	3	4	**3**	2	1	**2**	3	4	**3**	2	3	**4**	5	4	**3**	2	1	**2**	3	2	**1**	0

Fig. 16. The distances of bold binary values to other values

5.1 The Bold Binary Values

The bold binary values in the sequence is used as the values of clusters. In an artificial intelligence approach, an intelligent system must first be trained so, that it will memorize patterns. Having such memory makes the system be capable of for instance recognizing previously trained patterns. The memorized values of those for-training patterns will be clustered into those bold values. In this research, it will be investigated, how many clusters are the best configuration that will achieve the most accurate results. Now, whenever a pattern needs to be recognized by the mHGN, all the values bold or not will be recognized as bold values. For instance, if the value taken from particular sensor is 10001 (index 19), the mHGN will recognize it as 10000 (index 20). The reason to this is because the smallest Hamming Distance of the value 10001 happens to be between this value and the value 10000, which is 1. Similarly, if the value taken from particular sensor is 10010 (index 21), the mHGN will recognize it as 10000 (index 20) as well. The reason to this is because the smallest Hamming Distance of the value 10010 happens to be between this value and the value 10000, which is 1.

Someone might raise a question, whether there is a chance that the Hamming Distance 1 of the values 10001 and 10010 is not only compared to the bold value 10000, but also to one or two other bold values? To answer this question, the following Hamming Distance table shows the answer.

In the above table, it can be seen that in any bold row or in any bold column, the Hamming Distance of 1 is located in two places only. It means that the Hamming Distance of 1 belongs to those on the first adjacent neighbor values.

6 Conclusion

From the experiment results it is shown that mHGN has the capability to recognize multidimensional patterns. For simulating a tornado forecast, we have presented results of up to 5D architecture. As already discussed in [15] and [9] there is no modification required if the architecture needs to be extended to bigger sizes of patterns. In the future this capability will be improved to the extent so, that multi oriented of multidimensional patterns will also be recognizable. At this stage it is also observed that mHGN still use a single cycle memorization and recall operation. The scheme still utilizes small response time that is insensitive to the increases in the number of stored patterns.

References

1. R. T. Cheng and R. E. Smith, "A Nowcast Model for Tides and Tidal Currents in San Francisco Bay, California," in Ocean Community Conference, Baltimore, USA, 1998.
2. J. H. Sorensen, "Hazard Warning Systems: Review of 20 Years of Progress," Natural Hazards Review, pp. 119-125, May 2000.
3. D. B. Kirschbaum, R. Adler, Y. Hong, S. Kumar, C. Peters-Lidard and A. Lerner-Lam, "Advances in landslide nowcasting: evaluation of a global and regional modeling approach," Environment Earth Science, vol. 1, no. 1, pp. 1-14, 2011.

55

4. S. Bellaire and B. Jamieson, "Nowcast with A Forecast – Snow Cover Simulations on Slopes," Journal of Something, vol. 1, no. 1, pp. 1-7, 2012.

5. P. Li and E. Lai, "Applications of Radar-based Nowcasting Techniques for Mesoscale Weather Forecasting in Hong Kong," Meteorological Applications, vol. 11, no. 1, pp. 253-264, 2004.

6. S. S. Roy, V. Lakshmanan, S. R. Bhowmik and S. Thampi, "Doppler Radar based Nowcasting of Cyclone Ogni," Journal of Earth System Science, vol. 119, no. 2, pp. 183-199, 2009.

7. Y. SuzukiI, Y. Michihiro and M. Honma, "Development of Strong Wind Nowcasting System," in International Symposium on Extreme Weather and Cities, Tokyo, Japan, 2013.

8. B. B. Nasution, R. W. Sembiring, B. V. Sundawa, Gunawan, A. Amelia, Ismael, H. Sunjaya, S. Alifuddin, M. Pardede, Junaidi, M. Syahruddin and Z. Lubis, "Forecasting Natural Disasters of Tornados using mHGN," in IFIP Advances in Information and Communication Technology, Berlin, Springer, 2017.

9. B. B. Nasution, "Towards Real Time Multidimensional Hierarchical Graph Neuron (mHGN)," in The 2nd International Conference on Computer and Information Sciences 2014 (ICCOINS 2014), Kuala Lumpur, Malaysia, 2014.

10. T. Winterrath and W. Rosenow, "The Radar- based Precipitation Nowcasting System RADVOR of Deutscher Wetterdienst for the support of meteorological and hydrological alert systems," in 8th European Conference on Radar in Meteorology and Hydrology, Germany, 2014.

11. Y. Kryvasheyeu, H. Chen, N. Obradovich, E. Moro, P. V. Hentenryck and J. Fowler, "Nowcasting Disaster Damage," arXiv preprint arXiv:1504.06827, 2015.

12. J. Ogasawara, K. Tanimoto, O. Imaichi and M. Yoshimoto, "Disaster Prevention and Response Support Solutions," Hitachi Review, vol. 63, no. 1, pp. 236-243, 2014.

13. D. J. Doong, L. Z. H. Chuang, L. C. Wu, Y. M. Fan, C. C. Kao and a. J. H. Wang, "Development of An Operational Coastal Flooding Early Warning System," Natural Hazards and Earth System Sciences, vol. 12, no. 1, p. 379–390, 2012.

14. B. B. Nasution, R. W. Sembiring, B. V. Sundawa, Gunawan, A. Amelia, Ismael, H. Sunjaya, S. Alifuddin, M. Pardede, Junaidi, M. Syahruddin and Z. Lubis, "Realtime Weather Forecasting using Multidimenssional Hierarchical Graph Neuron (mHGN)," in The 16th International Conference on Neural Networks (NN '15), Rome, Italy, 2015.

15. B. B. Nasution and A. I. Khan, "A Hierarchical Graph Neuron Scheme for Real-time Pattern Recognition," IEEE Transactions on Neural Networks, pp. 212-229, 2008.