ISSN 2278-3091
Volume 8, No.1.3, 2019
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse5681.32019.pdf
https://doi.org/10.30534/ijatcse/2019/5681.32019

Triangular Fuzzy Number Generator (TriGen)

Muhammad Shukri Che Lah[1]*, Nureize Arbaiy[2]

[1]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Malaysia, shukrichelah89@gmail.com
[2]Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia, Malaysia, nureize@uthm.edu.my

ABSTRACT

The Triangular Fuzzy Number Generator (TriGen) demonstrated in this paper is a software tool which is developed to generate symmetry Triangular Fuzzy Number (TFN). Preparing TFN manually for fuzzy system or analysis requires calculation effort. When the number of data increases, calculating TFN requires longer time to complete and may exposed to calculation error. Hence TriGen is developed to diminish computational difficulties and reduce processing time of calculating TFN. This tool provides a function for capturing input data and calculates the spread of TFN based on Percentage Error and Standard Deviation approaches. The tool also embeds the formulas that allow it to calculate the actual result. Technical problem solving is described by numerical examples.

Key words: Triangular Fuzzy Number, Spread, Data Preparation Tool, Standard Deviation

1. INTRODUCTION

Data preparation involves the process of cleaning, transforming, consolidation or manipulation of raw data. It is an important process to transform a data source into a format, quality and structure suitable for further analytic [1][2]. Data preparation is crucial as it prepares data and its content to the next process. It also ensures data that is ready for analysis is accurate and consistent to produce valid analytics result. Though data preparation is said to time-consuming and crucial process, however it is required to produce accurate, meaningful and clean data to leverage the analytics. Data preparation is needed as raw data may contain discrepancies, outliers or errors, and/or lack of attributes. Data preparation ensures data to be formatted sufficiently and adheres to specific set of rules [2]. In data mining and machine learning, various techniques of data pre-processing has been used [3] and yield the improvement in the analytic model's performance. A formal data preparation process allows less time to find and structure the data sufficiently. Additionally, it enhances the performance and accuracy of the analytic model.

Fuzzy systems application in industrial and scientific applications are widespread [3]–[10]. In a fuzzy system, data is presented in a fuzzy number form to exhibit the fuzziness or uncertainty of the domain case study [11], [12]. Fuzzy number are defined as a generalization of a real number (crisp and non-fuzzy) to a set of possible values, between 0 and 1 [13]. This weight is known as the membership function. Based on membership function, fuzzy number is built. Fuzzy data can be presented in a form of center value (crisp) with it spread. However, data that is collected from real-world application (non-fuzzy) is not literally in the fuzzy format. Organizing and preparing the non-fuzzy data to fuzzy data format should be done correctly to avoid mistakes. Thus, computational tool is necessary to assist fuzzy data preparation efficiently while saving computation time and prevent mistakes.

Fuzzy numbers is used to represent fuzziness play an important role in many applications [13]. Nevertheless, the main difficulty in the development of applications is the computational complexity [12], [14]. Apart from algorithm development, more attention also should place to computerize data preparation and transformation for fuzzy numbers. As a volume data is grown, computational efficiency is concerned. The use of manual calculation for fuzzy data preparation is tedious due to the requirement of considerable time and effort from the certain programming expertise [15]. Hence, Triangular Fuzzy Number Generator (TriGen) is developed to reduce the computation task of fuzzy data preparation and transformation. This tool allows constructions of symmetry triangular fuzzy number $\tilde{y}$ in a form of center c and spread α, $\tilde{y} = [c, \alpha]$. Furthermore, TriGen provide function to upload real number datasets as input and allow user to download the generated spread of TFN in comma-separated values (CSV) format. Accompanied in the tool is the function that generate spread based on Standard Deviation and Percentage Error approaches. Such approaches are optimized to determine symmetry triangular fuzzy numbers to obtain the interval parameters in fuzzy forms.

The remaining of the paper is organized as follows. Section 2 provides the concept and formulation of triangular fuzzy number. Section 3 explains the structure of application tool with its input, process algorithm and output. Section 4 presents the result and discussion. Finally, Section 5 provides the overall conclusions.

2. TRIANGULAR FUZZY NUMBER GENERATOR (TRIGEN) PROCESS ALGORITHM

Section 2.1 describes formulation and calculation of triangular fuzzy number [12], [16], [17]. While Section 2.2 explains the Triangular Fuzzy Number Generator (TriGen) algorithm.

2.1 Triangular Fuzzy Number

Let $\tilde{y}$ is a triangular fuzzy number with membership function:

$$\tilde{y} = m(x) = \begin{cases} \frac{x-a}{b-a}, & x \in [a,b] \\ \frac{c-x}{c-b}, & x \in [b,c] \\ 0, & x < a \text{ and } x > c \end{cases} \quad (1)$$

From Eq. (1), triangular fuzzy number is defined as

$$\tilde{y} = [\alpha_l, c, \alpha_r]. \quad (2)$$

From Eq. (2), a symmetry triangular fuzzy number has same spread, $\alpha_2 - \alpha_1 = \alpha_3 - \alpha_2$. $\tilde{y}$ is denoted as

$$\tilde{y} = [c, \alpha], \quad (3)$$

where c is center value and α is spread of triangular fuzzy number. $\tilde{y}$ is non-fuzzy number if $\alpha = 0$.

In this study, the spread of symmetry TFN is generate based on Standard Deviation (SD) and Percentage Error (PE) approaches. In various fuzzy system application, usually spread is determined by the human expert with experience to define the fuzziness or uncertainty in the data [18]. However, less effort has been made to determine the spread in building fuzzy number with absence of expert justification. Though, determination of spread is important and need to be defined before TFN is built. Thus, the two approaches of standard deviation and percentage error are proposed to define the spread in a systematic procedural step.

Standard deviation is defined as follows:

$$S = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}} \quad (4)$$

where S, is standard deviation for population, x_i is each values of the data, $\bar{x}$ is the mean of x_i and n is the number of data points.

Based on standard deviation approach, the standard deviation σ value in Equation (4) is used to determine the spread S. The symmetry TFN of $\tilde{y}_t^S$ is as follows:

$$\tilde{y}_t^S = [y_t - s, y_t, y_t + s] \quad (5)$$

where $\tilde{y}_t^S$ is a fuzzy time series data at time, t with TFN form with spread of standard deviation, S and y_t is a time series data at time, t ($t = 1,2,\dots,n$).

Figure 1 shows the symmetry triangular fuzzy number based on Equation (5).

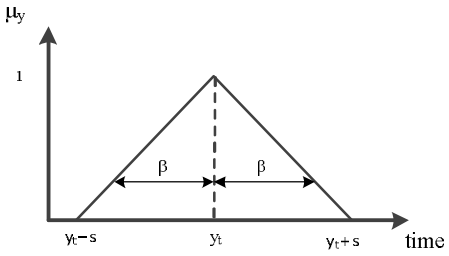


Figure 1: Symmetry TFN using standard deviation approach

While, in percentage error approach, the 95% confidence interval is optimized whereby the spread of TFN is adjusted from 5%, 3% and 1%. Only 5% spread and less are consider as the best spread to be used following the nature of confidence interval [19]. The symmetry TFN of $\tilde{y}_t^p$ is written as:

$$\tilde{y}_t^p = [y_t - y_t.p,\ y_t, y_t + y_t.p] \quad (6)$$

where $\tilde{y}_t^p$ is a fuzzy time series data at time, t with TFN form with spread, p ($p = 0.01, 0.03, 0.05$). y_t is a time series data at time, t ($t = 1,2,\dots,n$).

Percent error refer to the different between measured value and the accepted value. It is often used to report the difference between experimental values and accepted value.

$$percent\ error = \frac{|ev - av|}{av} \times 100\% \quad (7)$$

where ev is experimental value and av is accepted value.

In this paper, the TriGen tool is developed based on first order of autoregressive (AR1) model. The following process algorithm explains the steps taken by the TriGen.

a. Triangular Fuzzy Number Generator (TriGen) Algorithm

The algorithm process of TriGen is described as follows:

Step 1. Data input preparation.
Input format to generate TFN is $y_t = y_1, y_2, \dots, y_n$

Step 2. Calculate Standard Deviation
The Standard Deviation, σ is calculated according to the Equation (4).

Step 3: Calculate spread of Standard Deviation, S.

The spread of Standard Deviation, S of each data is calculated based on Equation (5).

Step 4. Calculate spread of Percentage Error, p.
The spread of Percentage Error, p of each data is calculated based on Equation (6).

Step 5. Generate symmetry triangular fuzzy number.
TFN are generated based on spread of Percentage Error, S and spread of Standard Deviation, p. Triangular fuzzy number will be computed by TriGen.

The existing TriGen is developed to provide computation for generating TFN. Practically, the tool is able to compute correctly, reduce computational difficulties and also reduce the computational time as compared to manual calculation by hand. User may opt for percentage error or standard deviation approach to define the spread for symmetry triangular fuzzy number. This tool is also capable to import AR(1) input through comma-separated values (CSV) format and transforming the input into TFN form.

3. THE SOFTWARE TOOL: TRIANGULAR FUZZY NUMBER GENERATOR (TRIGEN)

TriGen is built based on Javascript to generate the spread of TFN. While, HTML is used to design the interface of TriGen and with some element of CSS. The input, output and the main structure of the software is described in this section. The features provided by TriGen follows algorithm procedures which is defined in Section 2.2.

3.1. Input: Data Preparation

Time series datasets will be used as input to generate spread of TFN in this study. The dataset is sorted by the previous data to the most recent data. Dataset should be arranged in CSV file format. The input data format is as shown in Figure 2.

	A
1	6339.24
2	6251.48
3	6292.32
4	6353.74
5	6385.4
6	6373.14
7	6371.17
8	6386.34
9	6370.06
10	6382.19
11	6429.69
12	6444.52
13	6472.67
14	6490.9

Figure 2: Input datasets in CSV format.

The dataset which is uploaded to the TriGen is used to compute the TFN's spread. Figure 3 shows user interface for dataset file selection and upload function in TriGen.

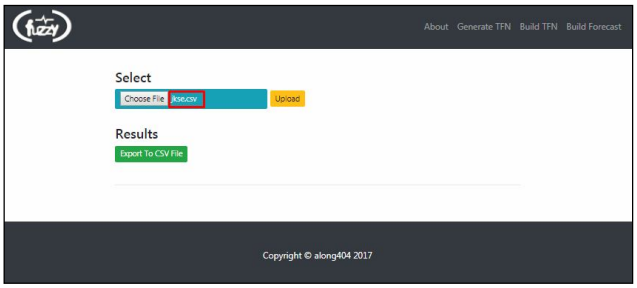


Figure 3: Dataset selection and upload function.

3.2. Output: The spread of TFN

The computation results are presented in the same view after the 'Upload' button is clicked. In this view we can see the generated spread of TFN. The calculation is computed based on the algorithm described in Section 2.2.

Functions is used to help the calculation and to provide an organized layout. Figure 4 describes the algorithm used to compile the entire process.

```
Step 1: START
Step 2: Read data from uploaded dataset
Step 3: Find Maximum number in dataset (Max)
Step 4: Find Standard Deviation for the dataset
Step 5: LOOP i from 0 to Max
        5.1: Generate spread of 1% Percentage Error
        5.2: Generate spread of 3% Percentage Error
        5.3: Generate spread of 5% Percentage Error
Step 6: Next i (back to Step 5 - until i <> Max)
Step 7: Loop i from 0 to Max
        7.1: Generate spread of Standard Deviation
Step 8: Next i (back to Step 5 - until i <> Max)
Step 9: END
```

Figure 4: Algorithm on the Structure of Data Flow.

The spread of triangular fuzzy number is computed by TriGen based on Standard Deviation and Percentage Error approach. The algorithm in Figure 4 is transformed into program code. Declaration is the main part in code and is demonstrated in Figure 5.

```
var table = $("#resultTable");
var resultHtml2 = '';
resultHtml2 += ["<tr>",
    '<th> </th>',
    '<th><font size="3"> y</font><font size="1">t</font></th>',
    '<th width="10%"><font size="3"> TFN</font><font size="1"> C</font></th>',
    '<th><font size="3"> TFN</font><font size="1"> 5% L</font></th>',
    '<th><font size="3"> TFN</font><font size="1"> 5% R</font></th>',
    '<th><font size="3"> TFN</font><font size="1"> 3% L</font></th>',
    '<th><font size="3"> TFN</font><font size="1"> 3% R</font></th>',
    '<th><font size="3"> TFN</font><font size="1"> 1% L</font></th>',
    '<th><font size="3"> TFN</font><font size="1"> 1% R</font></th>',
    '<th><font size="3"> TFN</font><font size="1"> Std Dev L</font></th>',
    '<th><font size="3"> TFN</font><font size="1"> Std Dev R</font></th>',
 "</tr>"];

var y = 0;
var length_csv = 0;
var keep_all = new Array();
var keep_all_yt = new Array();
```

Figure 5: Variable declaration.

Another important part in coding is the process. It is the most critical part because how the programmer organizes the code will influence the performance of the developed tool. Figure 6 shows code segment to read the input and keep the data in the array.

```
var regex = /^([a-zA-Z0-9\s_\\.\-:])+(.csv|.txt)$/;
if (regex.test($("#fileUpload").val().toLowerCase())) {
    if (typeof (FileReader) != "undefined") {
        var reader = new FileReader();
        reader.onload = function (e) {
            var rows = e.target.result.split("\n");
            length_csv = rows.length;
            for (var i = 0; i < length_csv -1; i++) {

                    keep_all[i]=rows[i];

                    var yt = rows[i];

                    keep_all_yt[i]=yt;
                }
```

Figure 6: Data arrangement.

The array of data is then used to generate the spread of TFN with related Equation from Section 2.1. Figure 7 shows the code segment to generate the spread of TFN.

```
for (var i = 0; i < length_csv-1; i++) {

    yt = keep_all_yt[i];

    var yt5 = yt * 0.05;
    var yt3 = yt * 0.03;
    var yt1 = yt * 0.01;
```

Figure 7: TFN spread generator.

```
function standardDeviation(arr){
  var n = arr.length;
    var sum = 0;

    arr.map(function(data) {
        sum+=Number(data);
    });

    var mean = Number(sum) / Number(n);

    var variance = 0;
    var v1 = 0;
    var v2 = 0;

    if (n != 1) {
        for (var i = 0; i<n; i++) {
            v1 = v1 + (arr[i] - mean) * (arr[i] - mean);
            v2 = v2 + (arr[i] - mean);
        }

        v2 = v2 * v2 / n;
        variance = (v1 - v2) / (n-1);
        if (variance < 0) { variance = 0; }
        stddev = Math.sqrt(variance);
    }
    return stddev;

}
```

Figure 8: Standard Deviation calculation.

There is also special function that has been set to calculate the value of the standard deviation which is based on Population Standard Deviation. Figure 8 shows the function to calculate the Standard Deviation. Figure 9 shows the program code to display the result, which is generated spread of TFN.

```
resultHtml2 += [
'<tr id="row1">',
    "<td>", "yt" + (i+1), "</td>",
    '<td>',yt,'</td>',
    '<td>',yt,'</td>',
    "<td>",(yt-yt5).toFixed(4),"</td>",
    "<td>",(Number(yt)+Number(yt5)).toFixed(4),"</td>",
    "<td>",(yt-yt3).toFixed(4),"</td>",
    "<td>",(Number(yt)+Number(yt3)).toFixed(4),"</td>",
    "<td>",(yt-yt1).toFixed(4),"</td>",
    "<td>",(Number(yt)+Number(yt1)).toFixed(4),"</td>",
    "<td>",(yt-stdDev).toFixed(4),"</td>",
    "<td>",(Number(yt)+Number(stdDev)).toFixed(4),"</td>",
'</tr>'].join("\n");
```

Figure 9: Algorithm on the Structure of Data Flow.

The download function is provided for user to transfer the generated spread of TFN into CSV file format for further analysis. Figure 10 shows the code segment for data export.

```
function exportTableToCSV(filename) {
  var csv = [];
  var rows = document.querySelectorAll("table tr");

  for (var i = 0; i < rows.length; i++) {
      var row = [], cols = rows[i].querySelectorAll("td, th");

      for (var j = 0; j < cols.length; j++)
          row.push(cols[j].innerText);

      csv.push(row.join(","));
  }

  // Download CSV file
  downloadCSV(csv.join("\n"), filename);
```

Figure 10: Export and download function.

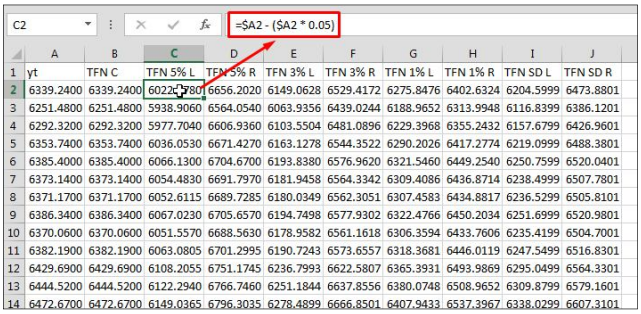
Figure 11 shows the code which is used for TriGen interface to display all the generated TFN's spread.

```
<section id="about" style=" padding-top: 50px; padding-bottom: 50px;">
  <div class="container">
    <div class="row">
      <div class="col-lg-8 mx-auto">
        <div>
                <h4>Select</h4>
                <input type="file" class="btn btn-info btn-sm" id="fileUpload" />
                <input type="button" class="btn btn-warning btn-sm" id="upload" value="Upload" />
                <hr />
        </div>

        <div id="table-gen">
            <h4>Results</h4>
            <div>
                <button onclick="exportTableToCSV('TFN_Data.csv')" class="btn btn-success btn-sm">Export To CSV File</button>
            </div>
            <br>
            <table id="resultTable">
                <thead>

                </thead>
            </table>
        </div>
        <hr />
      </div>
    </div>
  </div>
</section>
```

Figure 11: Result displays.

4. RESULTS AND DISCUSSION

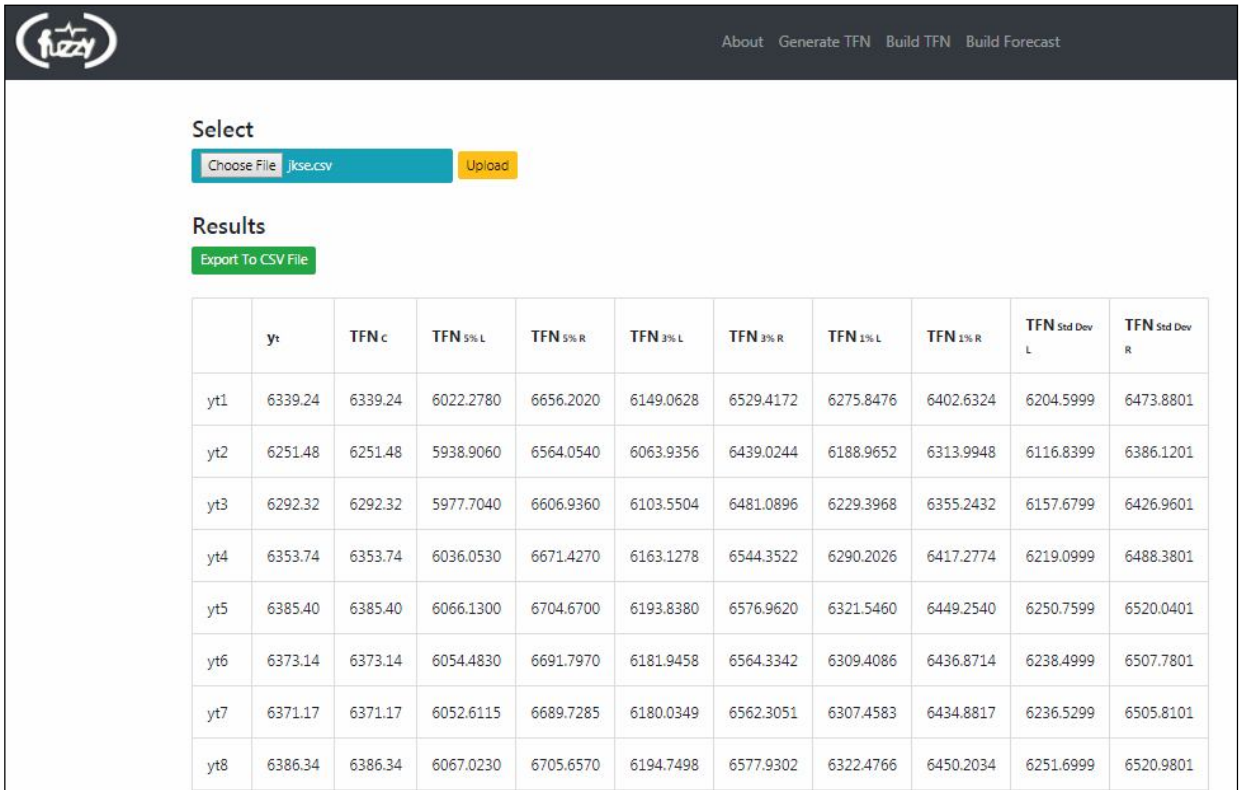
Based on the algorithm process in Section 2.2, TriGen produces the spread of TFN from single point data (input). The generated result computed by TriGen yield similar values with manual calculation by using Microsoft Excel©. In contrast, user need to set each column with percentage error or standard deviation formula, as shown in Figure 12. However, in TriGen such calculation is automatically completed by the tool.



	A	B	C	D	E	F	G	H	I	J
1	yt	TFN C	TFN 5% L	TFN 5% R	TFN 3% L	TFN 3% R	TFN 1% L	TFN 1% R	TFN SD L	TFN SD R
2	6339.2400	6339.2400	6022.2780	6656.2020	6149.0628	6529.4172	6275.8476	6402.6324	6204.5999	6473.8801
3	6251.4800	6251.4800	5938.9060	6564.0540	6063.9356	6439.0244	6188.9652	6313.9948	6116.8399	6386.1201
4	6292.3200	6292.3200	5977.7040	6606.9360	6103.5504	6481.0896	6229.3968	6355.2432	6157.6799	6426.9601
5	6353.7400	6353.7400	6036.0530	6671.4270	6163.1278	6544.3522	6290.2026	6417.2774	6219.0999	6488.3801
6	6385.4000	6385.4000	6066.1300	6704.6700	6193.8380	6576.9620	6321.5460	6449.2540	6250.7599	6520.0401
7	6373.1400	6373.1400	6054.4830	6691.7970	6181.9458	6564.3342	6309.4086	6436.8714	6238.4999	6507.7801
8	6371.1700	6371.1700	6052.6115	6689.7285	6180.0349	6562.3051	6307.4583	6434.8817	6236.5299	6505.8101
9	6386.3400	6386.3400	6067.0230	6705.6570	6194.7498	6577.9302	6322.4766	6450.2034	6251.6999	6520.9801
10	6370.0600	6370.0600	6051.5570	6688.5630	6178.9582	6561.1618	6306.3594	6433.7606	6235.4199	6504.7001
11	6382.1900	6382.1900	6063.0805	6701.2995	6190.7243	6573.6557	6318.3681	6446.0119	6247.5499	6516.8301
12	6429.6900	6429.6900	6108.2055	6751.1745	6236.7993	6622.5807	6365.3931	6493.9869	6295.0499	6564.3301
13	6444.5200	6444.5200	6122.2940	6766.7460	6251.1844	6637.8556	6380.0748	6508.9652	6309.8799	6579.1601
14	6472.6700	6472.6700	6149.0365	6796.3035	6278.4899	6666.8501	6407.9433	6537.3967	6338.0299	6607.3101

Figure 12: Construction of TFN in Microsoft Excel©.

TriGen provide a file upload function. User need to prepare datasets such as illustrated in Figure 1, and upload into TriGen. The tool will automatically generate spread of TFN based on Standard Deviation and Percentage Error approach. Figure 13 shows the generated spread of TFN. In addition, TriGen provide function that allow user to download the spreads in



	y_t	TFN c	TFN 5% L	TFN 5% R	TFN 3% L	TFN 3% R	TFN 1% L	TFN 1% R	TFN Std Dev L	TFN Std Dev R
yt1	6339.24	6339.24	6022.2780	6656.2020	6149.0628	6529.4172	6275.8476	6402.6324	6204.5999	6473.8801
yt2	6251.48	6251.48	5938.9060	6564.0540	6063.9356	6439.0244	6188.9652	6313.9948	6116.8399	6386.1201
yt3	6292.32	6292.32	5977.7040	6606.9360	6103.5504	6481.0896	6229.3968	6355.2432	6157.6799	6426.9601
yt4	6353.74	6353.74	6036.0530	6671.4270	6163.1278	6544.3522	6290.2026	6417.2774	6219.0999	6488.3801
yt5	6385.40	6385.40	6066.1300	6704.6700	6193.8380	6576.9620	6321.5460	6449.2540	6250.7599	6520.0401
yt6	6373.14	6373.14	6054.4830	6691.7970	6181.9458	6564.3342	6309.4086	6436.8714	6238.4999	6507.7801
yt7	6371.17	6371.17	6052.6115	6689.7285	6180.0349	6562.3051	6307.4583	6434.8817	6236.5299	6505.8101
yt8	6386.34	6386.34	6067.0230	6705.6570	6194.7498	6577.9302	6322.4766	6450.2034	6251.6999	6520.9801

Figure 13: Generated spread of TFN using TriGen.

	A	B	C	D	E	F	G	H	I	J
1	yt	TFN C	TFN 5% L	TFN 5% R	TFN 3% L	TFN 3% R	TFN 1% L	TFN 1% R	TFN SD L	TFN SD R
2	6339.2400	6339.2400	6022.2780	6656.2020	6149.0628	6529.4172	6275.8476	6402.6324	6204.5999	6473.8801
3	6251.4800	6251.4800	5938.9060	6564.0540	6063.9356	6439.0244	6188.9652	6313.9948	6116.8399	6386.1201
4	6292.3200	6292.3200	5977.7040	6606.9360	6103.5504	6481.0896	6229.3968	6355.2432	6157.6799	6426.9601
5	6353.7400	6353.7400	6036.0530	6671.4270	6163.1278	6544.3522	6290.2026	6417.2774	6219.0999	6488.3801
6	6385.4000	6385.4000	6066.1300	6704.6700	6193.8380	6576.9620	6321.5460	6449.2540	6250.7599	6520.0401
7	6373.1400	6373.1400	6054.4830	6691.7970	6181.9458	6564.3342	6309.4086	6436.8714	6238.4999	6507.7801
8	6371.1700	6371.1700	6052.6115	6689.7285	6180.0349	6562.3051	6307.4583	6434.8817	6236.5299	6505.8101
9	6386.3400	6386.3400	6067.0230	6705.6570	6194.7498	6577.9302	6322.4766	6450.2034	6251.6999	6520.9801
10	6370.0600	6370.0600	6051.5570	6688.5630	6178.9582	6561.1618	6306.3594	6433.7606	6235.4199	6504.7001
11	6382.1900	6382.1900	6063.0805	6701.2995	6190.7243	6573.6557	6318.3681	6446.0119	6247.5499	6516.8301
12	6429.6900	6429.6900	6108.2055	6751.1745	6236.7993	6622.5807	6365.3931	6493.9869	6295.0499	6564.3301
13	6444.5200	6444.5200	6122.2940	6766.7460	6251.1844	6637.8556	6380.0748	6508.9652	6309.8799	6579.1601
14	6472.6700	6472.6700	6149.0365	6796.3035	6278.4899	6666.8501	6407.9433	6537.3967	6338.0299	6607.3101

Figure 14: Generated spread of TFN in CSV using Microsoft Excel©.

CSV format for further analysis. Figure 14 show the generated spreads in CSV format.

Based on Figure 12 (manual) and Figure 14 (TriGen), both methods produce the same result. To validate the generated spread result with the manual calculation, we test the percent error for each spread based on Equation 7. This is to ensure that this tools produce the significant result with manual calculation.

Table 1: Percent error for generated spread.

5% PE		3% PE		1% PE		SD	
Left	Right	Left	Right	Left	Right	Left	Right
0	0	0	0	0	0	0	0
* The percent error is in percentage (%)							

Table 1 shows the percent error for generated spread. From Table 1, we can see that the percent error for all spread is 0% which tell us that this tools have achieved the same result with the manual calculation. However, TriGen offer least computation effort in constructing TFN. Besides that, TriGen also reduce computational error compare to manual calculation because the formula is embedded inside the TriGen. This tool also provides a function to determine the spread.

5. CONCLUSION

This paper presents TriGen, a tool to build symmetry triangular fuzzy number. This tool accepts the single point input which will then be used to generate spreads of TFN. TriGen provides four variations spreads of TFN as a result from the standard deviation and percentage error approaches. From this result, extensive study and analysis can be conducted to select best spread to be used. Also, TriGen is able to compute the spread of TFN efficiently and uses less of computation time. This tool can also give users the option to download the generated spreads in CSV format. For enhancement, the function of TriGen will be embedded with first order autoregressive function. The tool should be able to predict using the generated spreads and produce the Means Square Error automatically.

ACKNOWLEDGEMENT

The author would like to extend its appreciation to the Ministry of Higher Education (MOHE) and Universiti Tun Hussein Onn Malaysia (UTHM). This research is supported by Tier 1 grant (Vot U895) and GPPS grant (Vot U975). The author thanks to the anonymous viewers for the feedback. The author also would like to thank Universiti Tun Hussien Onn Malaysia for supporting this research.

REFERENCES

[1] M. Kantardzic, **Data mining: concepts, models, methods, and algorithms**. 2011. https://doi.org/10.1002/9781118029145

[2] S. García, J. Luengo, and F. Herrera, **Data Preprocessing in Data Mining**, vol. 72. 2015. https://doi.org/10.1007/978-3-319-10247-4

[3] Hung T. Nguyen and M. Sugeno, **Fuzzy Systems: Modeling and Control**. Springer Science & Business Media, 2012.

[4] R. R. Yager and L. A. Zadeh, **An Introduction to Fuzzy Logic Applications in Intelligent Systems**. 2012.

[5] Y. Jin, **Advanced Fuzzy Systems Design and Applications**. 2003. https://doi.org/10.1007/978-3-7908-1771-3

[6] P. Angelov, **Evolving Intelligent Systems Methodology and Applications**. 2010. https://doi.org/10.1002/9780470569962

[7] A. Nureize and J. Watada, "**A fuzzy regression approach to a hierarchical evaluation model for oil palm fruit grading**," Fuzzy Optim. Decis. Mak., vol. 9, no. 1, pp. 105–122, 2010. https://doi.org/10.1007/s10700-010-9072-3

[8] A. Nureize and J. Watada, "**Multi-attribute decision making in contractor selection under hybrid uncertainty**," J. Adv. Comput. Intell. Intell. Informatics, vol. 15, no. 4, pp. 465–472, 2011. https://doi.org/10.20965/jaciii.2011.p0465

[9] N. Arbaiy and J. Watada, "**Fuzzy goal programming for multi-level multi-objective problem: An additive model**," Commun. Comput. Inf. Sci., vol. 180 CCIS, no. PART 2, pp. 81–95, 2011. https://doi.org/10.1007/978-3-642-22191-0_7

[10] P. C. Lin and A. Nureize, "Two-Echelon Logistic Model Based on Game Theory with Fuzzy Variable," pp. 325–334, 2014. https://doi.org/10.1007/978-3-319-07692-8_31

[11] R. Viertl, **Statistical Methods for Fuzzy Data**. 2011. https://doi.org/10.1002/9780470974414

[12] D. Dubois and H. Prade, **Fundamentals of Fuzzy Sets**. The Handbook of Fuzzy Sets Series. 2000. https://doi.org/10.1007/978-1-4615-4429-6

[13] L. S. Senthilkumar, "**Triangular Approximation of fuzzy numbers - a new approach**," vol. 113, no. 13, pp. 115–121, 2017.

[14] and A. A. Sangaiah, Arun Kumar, Xiao-Zhi Gao, **Handbook of Research on Fuzzy and Rough Set Theory in Organizational Decision Making**. 2017. https://doi.org/10.4018/978-1-5225-1008-6

[15] N. P. Xiang, A. Nureize, H. A. Rahman, M. Zaki, and M. Salikon, "**Confidence Interval Calculator for Fuzzy Random Regression**."

[16] D. Dubois and H. Prade, "**Operations on fuzzy numbers**," Int. J. Syst. Sci., vol. 9, no. 6, pp. 613–626, 1978. https://doi.org/10.1080/00207727808941724

[17] H. T. Nguyen, "**A note on the extension principle for fuzzy sets**," J. Math. Anal. Appl., vol. 64, no. 2, pp. 369–380, 1978. https://doi.org/10.1016/0022-247X(78)90045-8

[18] J. Kacprzyk, H. Nurmi, and M. Fedrizzi, **Consensus Under Fuzziness**. 1997. https://doi.org/10.1007/978-1-4615-6333-4

[19] L. E. E. Schruben, **Confidence Interval Estimation Using Standardized Time Series**, vol. 31, no. 6. 1983. https://doi.org/10.1287/opre.31.6.1090