**Probing Time Series Multiplicative Model and Fuzzy Relation Methodologies**

**¹Heni I. Pratiwi, ²Yaya Heriadi, ³Lukas, ⁴Antoni Wibowo**

² ⁴Doctoral Program, Department of Computer Science, Bina Nusantara University, Jl. Anggrek, Kemanggisan, Jakarta, Indonesia ³Cognitive Engineering Research Group (CERG), Faculty of Engineering Universitas Katolik Indonesia Atma Jaya Jakarta, Indonesia

heni.pratiwi001@binus.ac.id, yaya.heriadi@binus.ac.id, lukas@atmajaya.ac.id, anwibowo@binus.edu

**Abstract.** Probing of a question between two methodologies of the multiplicative model of Time Series and Fuzzy Relation is challenging to answer, without exploring each of both theories. This paper aims to discern any discrepancies or similarities in them, since each has detailed items that could lead to practical advantages to proceed on forecasting or decision making. Pseudocoding is a chance to explore theories and to surface those practical advantages. The finding states that the multiplicative model of time series factually has components that could have dependencies one to another, and the fuzzy relations help to map each point along their processes to determine how much dependency contained in the system that could conclude of a decision making.

* **Introduction**

Definition of time series is a series of data points sequentially and measured successively within the allotted time [2]. Within the scope of the mathematical field, this measured time is defined as a set of vectors with the variable 't' to represent points of time that passed [2, 6]. Real modern lifestyle activities live undeniably in an optimistic cycle and involve a need for a future designed plan in time series model. The future designed plan is a forecasting focal goal with effective and optimum output.

Referring to the definition of time series, an application of the time series model, with time interval provisions that has been determined, has a variable x (t) could be set as a random variable [2 7]. Whether the time series is continuous or discrete, the determination of the specified time interval helps focus the observation and accuracy shall be achieved in chronological manner.

Recording temperature, velocity, stream flow, concentration of chemical components, wind direction, water pressure or electrical currents are some univariate observation examples in a continuous time series. Having the separation of time, in units of years, months, weeks, days, hours, minutes and seconds, is how discrete time series supports in observation output. The separation of time creates time intervals variables, and this is an example of a multivariate observation[6].

Along the observation to one or more variables in the time series tend to rise, fall and stagnate in different time intervals, and surface four main components: trend, cyclical, seasonal and irregular[7] These components are categorized according to the scale of required time distances,in other words, those categories for the component of trend requires a longer time frame scale,for example, the growth number of buildings in a city, a population, a state income, death rates due to epidemics, etc.The cyclical and seasonal components are categorized into shorter time frames than the trend components,for example, weather changes, religious and traditional activities, specific garment fashions according to summer or winter seasons, cold and flu seasons, etc. Natural disasters and incidents, political revolutions are some examples of a time frame for irregular components.

There are two types of models, in time series, they are called the multiplicative model and the additive model [6]. Multiplicative model is used when the basic assumption is having one or more of the four components considered independent, or dependent, and having factors that affect other components of the four. Additive model is used when the four components do not influence each other, in other words, each component is independent or stands alone.

In this paper the multiplicative model is referenced for the reasons of having assumptions that the forecasting design has a combined components of independent and dependent, so that additional tools are needed, in this case is the fuzzy relations. A group of objects could have an association with one another, the association creates a measurable relationship. This association relationship functions to supervise the mechanism of an interaction model and dependency between components, variable modules, etc.

Fuzzy relations have close interactions and operations such as in the Fuzzy set [1], which obviously has fundamental rules like in a set. Fuzzy relations applications are often found in the areas of pattern classification, diagnostics, modeling and control, decision-making, and information retrieval [3 ]. The fundamental rule in fuzzy relations is in a concept of relations by admitting the notion of partial association between elements of domains [nn], and this condition fits on what the multiplicative model in the time series needs.

Many research studies have been carried out with a focus on discussion about the size of uncertainty at the crisp relationship,for instance, a fuzzy relation under equivalent conditions, a fuzzy relation under similarity conditions and other crisp relations. The reason for writing this paper is to try complementing a forecasting method by combining multiplicative time series and fuzzy relations, while the outcome expectation could state that the element of uncertainty, contained in the multiplicative time series component, offers one or more justifications or solutions that is supported by fuzzy relations.

* **Methodologies**
* *Multiplicative Time Series*

Time Series as a set of vectors x(t), t = 0,1,2,... with ‘t’ represents the time elapsed[2 6 7]. The variable x(t) is treated as a random variable. Measurement is arranged in a proper chronological order, here the word proper means it is set as needed. Two different types of models are Multiplicative and additive models. Since multiplicative is reasoned to go with this paper, then the additive model is excluded from the discussion.

Multiplicative model is stated as Y(t) = T(t) x S(t) x C(t) x I(t). Y(t) represents the observation output, T(t) represents the trend component, S(t) represents the seasonal component, C(t) represents the cyclical component, I(t) represents the irregular component. Note that each component could be independent or dependent to one another, or could be a combination of dependent and independent, without necessarily involving all components.

* *Fuzzy Logic, fuzzy Set and Fuzzy Relations*

Fuzzy relation is a subtopic of fuzzy logic. In the beginning when the fuzzy logic was introduced by Zadech and continued by Mamdani, the main goal of fuzzy logic systems is to resolve complex processes by measuring what humans experienced [1]. Straightforward definition of fuzzy set and fuzzy relation items were well described by Wang[5]:

Let U = {x1,x2,··· ,xn} be a universe and A(·) : U → [0,1] be a mapping function, A is called a fuzzy set on U; for any x ∈ U, A(x) is called the membership degree of x to A.

Let A and B be two fuzzy sets on U, then A is called a fuzzy subset of B if A(x) ≤ B(x) for any x ∈ U. Let U × U be the Cartesian product of U, any fuzzy subset of U × U is then called a fuzzy binary relation on U, denoted by R.

Deﬁnition 2.2.1. Let R1,R2 ∈ F(U × U), ∀xi, xj ∈ U,

 (1) R1 = R2 ⇔ R1(xi,xj) = R2(xi,xj);

 (2) R1 ⊆ R2 ⇔ R1(xi,xj) ≤ R2(xi,xj);

 (3) (R1 R2) (xi,xj) = min{R1(xi,xj), R2(xi,xj)};

 (4) (R1 R2) (xi,xj) = max{R1(xi,xj), R2(xi,xj)}.

Deﬁnition 2.2.2. Let R ∈ F(U × U),

 (1) If R(xi,xj) = 0 for any xi,xj ∈ U, R is called an empty relation on U.

 (2) For any xi,xj ∈ U, if xi = xj, R(xi,xj) = 1; Otherwise, R(xi,xj) = 0.

 Then, R is called the identical relation on U.

 (3) If R(xi,xi) = 1 for any xi ∈ U, R is called a reﬂexive relation on U.

 (4) If R(xi,xj) = 1 for any xi,xj ∈ U, R is called the universal relation on U.

* **Discussion and Analysis**

Basically the practical method of analyzing time series models is to sort time intervals with its length scale and parameters for preprocessing or estimation process on data supplied. These are the procedures to select the right model in time series[8]. In time series forecasting, previously done observation results are compiled and analyzed to develop a mathematical model that captures the process of generating and underlying data for the series [1], then a method is formatted to understand the nature of series and beneficial for forecasting and simulating the future.

Time series forecasting has important roles in many sectors. Mostly valuable and strategic decisions, and precautions acts are taken based on forecasting output. When there is no such model that could fit the need, the approach of time series forecasting comes in handy to resolve the lack of statistical pattern data. Observing the multiplicative time series structure, the output Y (t) comes from a multiplication of components T (t), S (t), C (t) and I (t). There are facts that they could shape themselves in a multilayer fashion, for example the structure of the T (t) component could shape a T (S (t)) or S (C (t)) or T (S (C (I (t))) structures. Using practical thinking, a trend or T(t) could influence or be influenced by seasonal S(t) and cyclical C(t) and vice versa. Moreover, there are irregularity factors caused by some historical occured, such as a pandemic or political revolution, etc. A stratified or multiple structural information is potentially missing of its sub-structures mappings, since the relation to other attribute types was not detected. This caused the measure of uncertainty for this particular relationship has not been fully soluble, in other words, the method does not support handling mixed attribute types data. With a reminding note that generalizations of specific relations produce general fuzzy relations. [5]

What made a strong reason for adding the fuzzy relation method is that it generates a base set of rules to satisfy the fuzzy inferences, which expand for more accurate and optimal output. Since one or more components could influence or be influenced by one or more other components, then questions arise: how much stronger they influence each other to create the level of dependency and affects data processing results. Furthermore, those questions aggregate ideas about how the combination of component dependencies with one another are clearly mapped. This is the reason why fuzzy inferences are defined as a method that interprets the values in the input vector and, based on some set of rules, then assigns values to the output vector [5]. If fuzzy relations and multiplicative time series are combined into y (t) = (x1 (t)) x ( x2 (t)) x ( x3 (t)) x ( x4 (t)) since U = (x1 (t), x2 (t), x3 (t), x4 (t)) and the conditional which states that one or more components can be dependent or independent with one or more other components, the diagram in Figure 1 helps to explain:



**Figure 1**: Dependencies fuzzified

The good news is that fuzzy logic or fuzzy relation is simple, adaptable, and easily to fit on any other methods. In some cases, the need to recalibrate the fuzzy relation method is simply adjusting without rewriting the existing fuzzy logic rules. The following is the algorithm structure proposed in this paper, when it is supposed mapping the dependency relations among components within a binary number level between 1 and 0, binary 1 represents the maximum level and binary 0 represents the minimum. Here is the pseudocode:

start

lowDependency = 0.05;

averageDependency = 0.49;

highDependency = 0.59;

dependencyRange = highDependency - lowDependency;

#Note that intervals are aggregated from those levelling numbers, and this is only a design matter, those range numbers could be adjusted as needed.

low\_speedProcess = 0; #the more dependent one component, the lower speed process

average\_speedProcess = 5; #medium dependencies among components

high\_speedProcess = 10; #the less dependent one component, the higher speed process

speedProcess\_Range = high\_speedProcess - low\_speedProcess;

#speed ratio should derive from here, speed ratio should be negative when high\_speedProcess variable is the denominator of the range.

incorrect\_input = 0; #users should be alert incorrect input to proceed

correct\_input = 10; #this is a good start to proceed

inputCorrectness\_Range = correct\_input - incorrect\_input #this is needed to support correct output

% if dependency is low or input is correct, then the speed process fast,

 else if dependency < medium , then the speed could still affordably go fast

overall\_dependency = (((averageDependency - lowDependency ) / (average\_speedProcess - low\_speedProcess))\*speedProcess\_Range+lowDependency)\*speedRatio

+ (1-speedRatio)\*(dependencyRange / inputCorrectness\_Range\* input(#could be correct or incorrect) + lowDependency);

#speed ratio should derive from speedProcess\_Range variable, speed ratio should be negative when high\_speedProcess variable is the denominator of the range.

% if dependency is medium or input is correct, then the speed process average,

 else if dependency > medium , then the speed could start going slow

overall\_dependency = averageDependency \* speedRatio + ( 1- speedRatio)\* ( dependencyRange / inputCorrectness\_Range \* input(#could be correct or incorrect) + lowDependency);

% if dependency is high or input is correct, then the speed process longer,

 else if dependency is high or input is incorrect, then the speed process longer and slow,

overall\_dependency = ((( highDependency - averageDependency ) / (high\_speedProcess - medium\_speedProcess )) \* (speedRatio - medium\_speedProcess ) + averageDependency ) \*speedRatio + ( 1- speedRatio) \* ( dependencyRange / inputCorrectness\_Range \* input(#could be correct or incorrect) + lowDependency);

end

* **Summary**

The algorithms sketched at the discussion section could have more comments and modifications or fashions as needed. Moreover, personal experiences could make it more efficient and precise, since it becomes more specific. Usually the less precise an algorithm when it applies to general cases, and this paper aims to probe for more specific ways to work on some cases that seem promising. However some piecewise methodology probes are not yet the best to expect, some further more cases claim to resolve in effective accuracy.

Three main issues remain on probing multiplicative time series with fuzzy relations and it has aggregated the fuzzy logic rules:

• If processing speed is low or input is incorrect, dependency is high.

• If processing speed is average, components dependency is in medium

• If processing speed is high or input is correct, dependency is low.

Development could be more fuzzified on how input is specified and defined as correct or incorrect, and how many components exactly to twist each other into low or medium or high level definitions. These all create intervals numbers which what time series in multiplicative models operate, that what this paper aims for.

* **References**

[1] Volosenco C 2020 Fuzzy Logic (Research Gate)

[2] Bergmeir C, Hyndman R J, and Koo B 2018 A Note On The Validity of Cross-Validator Evaluating Autoregressive Time Series Prediction. Computational Statistic and Data Analysis 120, 70-83

[3] Mei W 2019 Formalization of Fuzzy Control in Possibility Theory via Rule Extraction (IEEE Access Vol.7)

[4] Li P, Zhang X and Ge, et al. 2018 Uncertainty Measurement For a Fuzzy Relation Information System (IEEE) 1063-6706

[5] Wang C, Huang Y, Shao M, and Chen D 2018 Uncertainty Measures For General Fuzzy Relations (London: Elsevier )

[6] Pahuja R and Mamidala R 2018 Process Monitoring In Milling Unidirectional Composite Laminates Through Wavelet Analysis of Force Signals Procedia Manufacturing (26) 645–655

[7] Athanasopoulos G, Hyndman R J, Kourentzes N, and Petropoulos F 2017 Forecasting With Temporal Hierarchies (European Journal of Operational Research) 262(1), 60–74.

[8] Ismail S and Shab A 2014 Time Series Forecasting using Least Square Support Vector Machine For Canadian Lynx Data (Research Gate)

[9] Guo H, Pedrycz W, and Liu X 2018 Fuzzy Time Series Forecasting Based on Axiomatic Fuzzy Set Theory. Neural Comput Appl. https ://doi. org/10.1007/s0052 1-017-3325-9 (article in press)