

Review

A Review of Power System Fault Diagnosis with Spiking Neural P Systems

Yicen Liu ¹, Ying Chen ², Prithwineel Paul ³ , Songhai Fan ¹, Xiaomin Ma ¹ and Gexiang Zhang ^{3,*} 

¹ State Grid Sichuan Electric Power Research Institute, Chengdu 610041, China; liuyi2664@sina.com (Y.L.); fansonghai@126.com (S.F.); xiaomage913@163.com (X.M.)

² State Grid Sichuan Electric Power Company, Chengdu 610041, China; leobodhi@163.com

³ Chengdu Zhonglian Huarui Artificial Intelligence Technology Co., Ltd., Chengdu 610041, China; prithwineelpaul@gmail.com

* Correspondence: zhgxdylan@126.com

Abstract: With the advancement of technologies it is becoming imperative to have a stable, secure and uninterrupted supply of power to electronic systems as well as to ensure the identification of faults occurring in these systems quickly and efficiently in case of any accident. Spiking neural P system (SNPS) is a popular parallel distributed computing model. It is inspired by the structure and functioning of spiking neurons. It belongs to the category of neural-like P systems and is well-known as a branch of the third generation neural networks. SNPS and its variants can perform the task of fault diagnosis in power systems efficiently. In this paper, we provide a comprehensive survey of these models, which can perform the task of fault diagnosis in transformers, power transmission networks, traction power supply systems, metro traction power supply systems, and electric locomotive systems. Furthermore, we discuss the use of these models in fault section estimation of power systems, fault location identification in distribution network, and fault line detection. We also discuss a software tool which can perform the task of fault diagnosis automatically. Finally, we discuss future research lines related to this topic.

Keywords: membrane computing; spiking neural P system; power systems; fault diagnosis



Citation: Liu, Y.; Chen, Y.; Paul, P.; Fan, S.; Ma, X.; Zhang, G. A Review of Power System Fault Diagnosis with Spiking Neural P Systems. *Appl. Sci.* **2021**, *11*, 4376. <https://doi.org/10.3390/app11104376>

Academic Editor: Rocco Furferi

Received: 5 April 2021

Accepted: 7 May 2021

Published: 12 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the 21st century, electrical power systems are associated with the daily necessities of human beings. A network of electrical components is called an electrical power system and such systems help in the supply, transfer, and use of electrical power. However, any accident inside the power systems interrupts the supply of power. In order to perform the task of secure and stable supply of power, it is necessary for the electric devices to have efficient fault diagnosis methods which can identify the faults quickly as well as efficiently. Furthermore, whenever there is a fault in power system, the consoles in the dispatchers receive huge number of messages in a very short span of time from the SCADA (supervisor control and data control) systems. Very often the messages received from SCADA systems are incomplete and uncertain. Power systems are composed of a large number of generators, transmission lines, bus bars, and transformers. These aspects of power systems make very difficult the task of fault diagnosis in power systems. In recent years, the techniques from artificial intelligence have been used to perform the task of fault diagnosis in electric power systems. More specifically, many methods based on Bayesian networks [1,2], artificial neural networks [3,4], genetic algorithms [5], Petri nets [6–9], expert systems [10,11], fuzzy logic [7,8,12], multi-agent systems [13,14], optimization methods [5,15,16], information theory [17,18], cause-effect networks [19,20] have been introduced. P systems or membrane computing models are popular natural computing model. In recent years, many researchers have used different variants of P systems for fault diagnosis in power systems.

P system was introduced in 1998 by Gh. Păun [21]. P system is an abstract mathematical computing model which is inspired by the structure and functioning of the biological cells. The area of study of P systems is well-known as *Membrane Computing*. In P systems, the objects present inside the cells or membranes are represented by multiset of objects and the evolution processes happening inside these cells are represented by the rules which are applied in maximal parallel manner. P systems models are generally divided into three categories based on their structure, i.e., cell-like, tissue-like and neural-like.

Since the introduction of the first model, there has been significant advancement in the study of membrane computing [22]. Because of the inherent parallel and distributed architecture and different mechanisms (such as active membranes [23,24], membrane divisions [25], etc.) cell-like P systems can generate exponential amount of space and then using the space-time trade-off can solve many NP-hard, NP-complete, PSPACE-complete problems [24–28] in polynomial and linear time. These models also can characterize many complexity classes [29,30]. The computational power [25,31–33] of different variants of P systems along with their complexity aspects [30,33–35] and use of the parallel and distributed architecture of these models in solving computational hard problems [26,33] have been a popular direction of research. More specifically, uniform families of shallow P systems with active membranes and charges can characterize the complexity classes $P^{\#P}$ and $P_{||}^{\#P}$ [36]. Moreover, shallow non-confluent P systems can characterize the family PSPACE [34]. The complexity class $P_{||}^{\#P}$ can be characterized by monodirectional shallow chargeless P systems [37] with active membranes and minimal cooperation working in polynomial time. The complexity class $P_{||}^{\#P}$ consists of the class of problems which can be solved in polynomial time by deterministic Turing machines with a polynomial number of parallel queries to an oracle for a counting problem. These studies raise an important question, i. e., what kind of problems can be solved by these models when certain bounds are introduced on the space resources. In [35], Zandron addresses this question. Moreover, since P systems communicate with each other using rules, it is important to investigate communication complexity of the P systems [38].

Along with finding the analytic solutions for computationally hard problems using the concept of space-time trade-off, in recent years the study of constructing evolutionary algorithms by combining the structure and operations of membrane systems and capabilities of optimization algorithms has gained prominence [39]. Furthermore, approximate solutions of these problems are obtained by using the algorithms which are popularly known as “membrane algorithms”. In [40], the solution of an well-known NP -complete problem, i.e., Graph coloring problem (GCP) is obtained by using OLMS (one level membrane structure (OLMS) [41–43]) membrane algorithm with dynamic operators. Automatic design of P systems is also an attractive work and promising research direction [44–47].

Membrane computing models have a wide range of real-life applications [39,48–52] and the investigation more problems from different areas of real-life applications which can be solved by these models has drawn huge interest amongst many researchers around the global. One of the most interesting application of membrane computing models has been in the area of “Robotics”. In particular, the use of membrane computing models for designing of membrane controllers for single and multi-robot systems which further helps in navigation of the robot in unknown environments [48,53–55]. In recent years, many variants of P systems such as enzymatic numerical P systems, XP-colonies, etc. have been used in single and multiple-robot applications. The membrane controllers based on the membrane computing models are efficient and have comparable performance with respect to traditional robot controllers [48]. Image processing is another area where membrane computing models have been used extensively. Parallel distributed architecture along with the multisets present in the membranes which is useful for encoding of the information, make P systems a suitable model for dealing with digital images. Furthermore, these models can be used in image segmentation, skeletonization, etc. [49]. Moreover, SNPS models can perform binary operations and these models are also suitable to be used in cryptographic applications [56]. Array rewriting P systems can be used as a tool for

generation of Peano curve, the Hilbert curve, etc. In [57], a state-of-the-art based on the P systems with parallel rewriting was introduced for generation of the space-filling curves, and related curves.

Membrane computing models also have been used for modelling and simulation of many phenomena existing in Biochemistry, Ecology, Robotics or Engineering [51]. These models also can model communities of very simple reactive agents living and acting in a joint shared environment. These types of membrane computing models are known as “P colonies” [58]. P colonies can simulate the interactive processes in other mechanisms such as reaction system which is inspired from the different chemical reactions happening in the environment [59]. P systems are also useful in modeling many biological phenomena such as swarming and aggregating behaviour in Myxobacteria bacterial populations [60,61].

Neural-like P systems/Spiking neural P systems (SNPS) [62] have gained popularity amongst the researchers because of its similarities with third-generation neural networks, i.e., spiking neural networks (SNNs) [63]. In recent years, the research in SNPS also has gained huge momentum and many variants of SNPS have been introduced inspired by the properties present in the biological neurons. Many variants of it already have been introduced along with investigations regarding their computational power [64,65], efficiency in solving computationally hard problems [66] and real-life applications [67–69]. A new variant of SNPS is introduced in [32] where the firing and forgetting rules are applied in generalized manner. In these models, if a rule is applied at any step of the computation, then it will be applicable any number of times. These models are also computational complete. Matrix representation and simulation algorithm of different variants of SNPS are investigated in [70]. Another novel variant of SNPS is recently introduced in [71] and it is called as dynamic threshold neural P systems. These models are inspired from the spiking and dynamic threshold mechanisms of neurons. Moreover, it has been proved that the sequential variant of this model is capable of generating/accepting Turing universal numbers. Some of the well-known variants of SNPS are SNPS with asynchronous systems [72], astrocytes [73], rule on synapses [74], communication on request [75], synapses with schedules [76], structural plasticity [77,78], weighted synapses [79], inhibitory synapses [80], anti-spikes [81], etc. Furthermore, SNPS have been used extensively in solving many real-world problems in many areas such as fault diagnosis of power systems [82–91], pattern recognition [92–94], computational biology [95], performing arithmetic and logical operations and hardware implementation [96–103], biochip design [104], programming for logic controllers [105,106], etc. In [107], SNPS models have been used for computing finite-state functions.

In the last few decades, although significant progress has been made in obtaining many theoretical results as well as real-life applications of membrane computing models, very little progress has been made towards in vivo implementation of these models. In [50], a mechanism was introduced where multivesicular liposomes can be experimentally produced through electroformation of dipalmitoylphosphatidylcholine films. It can be further used in ‘real’ P-systems. This study initiates the idea of constructing computing devices based on P systems and also investigates the limitation of these models.

Over the years one of the most interesting direction of research in membrane computing has been constructing simulation tools for different variants of P systems and their applications [108]. Many researchers have focused on developing softwares [109] based on P-Lingua which can simulate different variants of membrane systems efficiently such as (1) Cell-like P systems [110]; (2) P-Systems with String Replication [111]; (3) Tissue P systems [112] (4) Cell-like SNPS [113]; (5) Spiking Neural P Systems [114]; (6) Asynchronous Spiking Neural P Systems [115]. P-Lingua is an efficient tool. Recently, there has been significant progress towards constructing P-Lingua language, pLinguaCore library and MeCoSim environment. These tools also have been used for experimentally validate solutions of computationally hard problems [116]. Furthermore, many simulators have been proposed which are based on different programming languages other than P-Lingua and following is the list of some simulators: (1) CuSNP [117]; (2) UPSimulator [118]; (3)

Psim [119]; (4) SNUPS [120]; (5) GPUPeP [121]. The P systems simulators work as an important tool for formally creating a framework for real-life applications. The simulators based on the requirements of the users and specific application abstract the concepts of different variants of P systems. The researchers in membrane computing community have constructed many types of simulators [122] and used it for simulation of the simple kernel P systems solution to the graph 3-colouring problem [123]. Moreover, testing methods for membrane computing models such as kernel P systems have been introduced [124].

The process of fault diagnosis in power systems is based on processing of uncertain and incomplete information. General SNPS models are not capable of handling these type of information. So a new variant of SNPS was introduced in order to serve this purpose. In [84], a new variant of SNPS, i.e., FRSNPS (fuzzy reasoning spiking neural P systems) was introduced, aiming to handle fuzzy diagnosis knowledge and reasoning. Furthermore, a methodology for fault diagnosis in transformers was presented. There has been only a few investigations regarding incorporating the idea of machine learning in SNPS. It is also difficult to introduce machine learning mechanisms in SNPS because of its formal language theory framework. Recently, a new method based on learning Spiking Neural P System with Belief AdaBoost [91] is introduced for fault diagnosis in transformers. Moreover, SNPS models and their variants have been introduced for identification of faults in power transmission networks [125], traction power supply systems of high-speed railways [88], metro traction power systems [82], electric locomotive systems [87]. Moreover, these models have been used for fault location estimation of power systems [126] and fault line detection [85]. In this paper, we have summarized all the SNPS models and their corresponding methodologies for fault diagnosis in different types of power systems. More specifically, we discuss the structure and working of different variants of SNPS and studied the important aspects of the reasoning algorithms based on these models. Moreover, we discuss the advantages and disadvantages of different fault diagnosis methods. At the end, we discussed an automatic implementation method for fault diagnosis [127].

The main motivations for preparing this paper are as follows:

- (1) Until now there has been no comprehensive survey collecting all the methodologies of fault diagnosis of power systems based on the membrane computing models;
- (2) Fault diagnosis of power systems with spiking neural P systems is an important application of membrane computing. This review aims to highlight and advance this direction.
- (3) One more motivation of this review is to advance a newly emerging research direction, i.e., learning spiking neural P systems, by discussing a supervised learning algorithm in the framework of SNPS and using this model for identification of faults in transformers.

Furthermore, the contributions of this paper can be summarized in the following manner:

- (1) This paper provides a comprehensive survey of SNPS models and a software tool which are used for fault diagnosis in power systems;
- (2) The comparisons of different fault diagnosis methods and reasoning algorithms are studied in this paper.
- (3) This paper discusses future research lines with respect to SNPS for power system fault diagnosis and learning SNPS.

This paper is organized in the following manner: Section 2 discusses fault diagnosis methods in power systems in the framework of SNPS and its variants. Section 3 discusses an automatic implementation method for fault diagnosis in complex power systems and finally in Section 4, we conclude the paper and discuss some future research lines.

2. Power System Fault Diagnosis with Spiking Neural P Systems

Power systems have become an integral part of the modern civilization and it consist of electronic components. It is important for stable and uninterrupted supply of power to identify the faults in the power systems whenever there is an accident. Over the years

many methods [1–5,7–20] have been introduced by many researchers for identification of faults in the power systems efficiently and quickly. However, the use of membrane computing models, i.e., spiking neural P systems for fault identification in power systems is a recent phenomena. In this section, we discuss these models. More specifically, we divide this section into the following subsections based on type of power systems and faults, i.e., (1) fault diagnosis for transformers; (2) fault diagnosis for power transmission networks; (3) fault diagnosis for traction power supply systems; (4) fault diagnosis in metro traction power systems; (5) fault section estimation of power systems; (6) fault location identification of distribution network; (7) fault lines detection; (8) fault diagnosis in electric locomotive systems.

2.1. Transformers

In this section we discuss the use of two variants of spiking neural P systems (SNPS) for identification of faults in power transformers. Along with parallel and distributed architecture, SNPS models have some inherent properties such as high understandability, non-linearity, non-determinism, dynamically adaptability etc. and these properties make SNPS an useful tool for identification of faults in power systems. We divide this section into two parts. In the first part, we discuss the use of the fuzzy reasoning spiking neural P systems (FRSNPS) models for fault diagnosis in power transformers. Next we discuss the use of the learning Spiking Neural P System (LSNPS) with Belief AdaBoost for the same purpose.

In [84], Peng et al. introduced a new variant of SNPS, i.e., FRSNPS for fault diagnosis in transformers based on dissolved and free gas analysis (DGA). FRSNPS model is an extended variant of SNPS model and it has some distinctive properties. Furthermore, new types of fuzzy production rules were introduced in the framework of FRSNPS according to the fuzzy production rules. Moreover, a fuzzy reasoning algorithm was introduced to identify the faults in the transformer. In fuzzy reasoning algorithm, IEC ratio of gases is given as input to FRSNPS. Next the fault reasoning results are obtained in the form of CF (confidence factor)/truth values and then the faults in the transformers are identified by observing the CFs of different types of faults.

A power transformer is an important component of power systems which works as transmission and transformation equipment. Whenever there is any fault in power transformers, the stable and continuous supply of power is interrupted. So it is imperative to identify the faults in the transformer efficiently and quickly. It is important to note that due to the decomposition of the insulating oil, hydrocarbon and hydrogens are produced in the transformer and it happens because of the factors such as thermal, electrical, ambient, mechanical (TEAM) [84]. Furthermore, the level of insulation and symptom of faults occurring in the transformer are indicated by the concentration of hydrocarbon and hydrogen. DGA (dissolved gas analysis) has long been considered as an efficient and effective method for identification of faults in power transformers. It is well-known that ethane (C_2H_6), ethylene (C_2H_4), acetylene (C_2H_2), hydrogen (H_2), methane (CH_4) gases are present as dissolved gas in transformer insulating oil. The state of the transformer can be obtained by analyzing the concentrations of specific dissolved gases in insulation oil. It further helps in taking preventive actions. Moreover, from the difference of the ratio of gas, the faults in the power transformer such as partial discharge and spark discharge can be obtained.

In what follows we explain the fault diagnosis process. We begin with the discussion about the structure and working of FRSNPS model.

Definition 1. A FRSN P system (FRSNPS) [84] of degree $m \geq 1$, is a $(m + 4)$ -tuple $\Pi = (O, \sigma_1, \dots, \sigma_m, syn, IN, OUT)$ where

- $O = \{a\}$ is the singleton alphabet (the object a is called spike);
- $\sigma_1, \dots, \sigma_m$ denotes m neurons of the form $\sigma_i = (\alpha_i, \tau_i, r_i)$ ($i \in \{1, 2, \dots, m\}$) where
 - (i) $\alpha_i \in [0, 1]$ represents the potential value/pulse value of spike contained in neuron σ_i ;
 - (ii) $\tau_i \in [0, 1]$ represents the truth value associated with neuron σ_i ;

- (iii) r_i represents the firing/spiking rule contained in neuron σ_i and it is of the form $E/a^\alpha \rightarrow a^\beta$, where $\alpha, \beta \in [0, 1]$.
- $\text{syn} \subseteq \{1, 2, \dots, m\} \times \{1, 2, \dots, m\}$ with $i \neq j$ for all $(i, j) \in \text{syn}$ (synaptic connections between the neurons);
 - IN and OUT represent input neuron set and output neuron set, respectively.

When FRSNP is used for fault diagnosis, fuzzy production rules and reasoning algorithms are the most important components. Fuzzy production rules help us to express the diagnosis knowledge which is extracted from real-world data while addressing the problem of fault diagnosis. Fuzzy production rules are generally of two types, i.e., simple and composite [84]. Simple fuzzy productions have the following form

$$R_i: \text{IF } p_j \text{ THEN } p_k (CF = \tau_j) \quad (1)$$

where $\tau_i \in [0, 1]$ represents CF and R_i is the i th fuzzy production rule, p_j and p_k are propositions with real truth values in $[0, 1]$.

The composite fuzzy production rules are generally divided into three types, i.e., type-1, type-2 and type-3.

$$\text{Type 1 } R_i : \text{IF } p_1 \text{ and } p_2 \text{ and } \dots \text{ and } p_{k-1} \text{ THEN } p_k (CF = \tau_i). \quad (2)$$

In this rule p_1, p_2, \dots, p_k are propositions which are present in the antecedent part where the truth value of the propositions are $\alpha_1, \alpha_2, \dots, \alpha_{k-1}$. The confidence factor of the rule is represented by $\tau_i \in [0, 1]$. So correspondingly $\alpha_k = \min(\alpha_1, \alpha_2, \dots, \alpha_{k-1}) * \tau_i$ represents the truth value of the proposition p_k .

$$\text{Type 2 } R_i : \text{IF } p_1 \text{ THEN } p_2 \text{ and } p_3 \text{ and } \dots \text{ and } p_k (CF = \tau_i). \quad (3)$$

In this rule p_1 is only one proposition present in the antecedent part of the rule with truth value α_1 and $\tau_i \in [0, 1]$ is the confidence factor. So correspondingly $\alpha_2 = \alpha_1 * \tau_i, \alpha_3 = \alpha_1 * \tau_i, \dots, \alpha_k = \alpha_1 * \tau_i$ are the truth values of p_2, p_3, \dots, p_k respectively.

$$\text{Type 3 } R_i : \text{IF } p_1 \text{ or } p_2 \text{ or } \dots \text{ or } p_{k-1} \text{ THEN } p_k (CF = \tau_i). \quad (4)$$

In this rule p_1, \dots, p_{k-1} are the propositions present in the antecedent part of the rule and they have truth values $\alpha_1, \alpha_2, \dots, \alpha_{k-1}$, respectively. $\tau_i \in [0, 1]$ represents the confidence factor of the rule. So, $\alpha_k = \max(\alpha_1, \alpha_2, \dots, \alpha_{k-1}) * \tau_i$ represents the truth value of the proposition p_k .

One of the most important aspect of the fault diagnosis models based on SNPS models is to reconstruct the fuzzy production rules using the framework of SNPS. In this case, the fuzzy production rules are reconstructed using the framework of FRSNPS. In order to perform this task, different types of neurons are required and the neurons in the FRSNPS model are divided into two types, i.e., proposition neuron and rule neuron. Moreover, not all the neurons are associated with truth values and these values are generally real numbers. Again, the truth values are associated with some of the proposition neurons before the reasoning algorithm is performed. The corresponding proposition neurons work as input neurons of the FRSNPS model where IEC ratios of gases are given as input. Moreover, the truth values of the neurons which do not have any value associated with them are derived using the known truth values and reasoning algorithm. The concepts of the firing mechanisms of the neurons and matrix operations are used in the reasoning algorithm introduced in [84]. Moreover, the proposed fault diagnosis model can efficiently model fuzzy production rules in a fuzzy diagnosis knowledge base and their reasoning process. It is also feasible and effective in identifying the faults in transformers. It can also handle incomplete and uncertain information received from the SCADA and efficiently process it in order to find faults. One of the major advantage of the proposed method is the graphical representation of the reasoning process which makes the reasoning process

easily comprehensible. Finally, the reasoning process is parallel. So it can obtain results in much lesser time in comparison with tradition fault diagnosis models.

Another fault diagnosis method of power transformer based on DGA and FRSNPS is introduced in [128]. The FRSNPS model and the reasoning algorithm used in [128] are same as [84]. In this model, the faults in transformers are identified using linguistic variables, membership functions with low, medium and high descriptions for each gas signature and the inference rule base. In fact, the linguistic terms in [128] are very low (VL), low (L), medium (M), high (H), very high (VH), very low (VL). Similarly, little high, low, rather high, too high, high, little low, rather low and too high are the linguistic terms in [84]. In [84], the faults are divided into mainly four types, i.e., General overheating fault occurs, Serious overheating fault occurs, The partial discharge occurs, The spark discharge occurs. Moreover, the faults are Low energy discharge, High energy discharge, Partial discharge, Thermal faults $T < 300$ °C, Thermal faults 300 °C $< T < 700$ °C, Thermal faults $T > 700$ °C. In order to identify these faults, at first the judgements are translated into numerical expression by fuzzy numbers with the help of membership functions. The fault types interpretation of DGA and linguistic terms (L.T) are different from the proposed model in [84]. Next, four gas ratio (IEC 60599) signature is given as input to the FRSNPS and performance of the method is analyzed based on this method. From the test case results, it is inferred that the proposed method in [128], significantly improves the diagnosis accuracy of the power transformer.

A novel method of identifying faults for oil-immersed power transformers using the framework of SNPS is recently introduced in [91]. In this method, a new variant of SNPS, i.e., learning spiking neural P system (LSNPS) is introduced to perform this task. More specifically, the fault diagnosis method in [91] is based on LSNPS with belief AdaBoost and it is used for fault diagnosis in oil-immersed power transformer.

Now in order to explain the diagnosis process, at first the structure and working of the LSNPS with AdaBoost model are explained. LSNPS models have a feedforward network structure and can be mathematically formulated in the following manner. Since the sets O, IN, OUT, syn represent the same as in [84]. So we will not explain them in detail.

Definition 2. A LSN P system (LSNPS) [91] is a construct $\Pi = (O, \{\sigma_{p1}^1, \dots, \sigma_{pk}^1, \sigma_{p1}^3, \sigma_{p1}^5\}, \{\sigma_{r1}^2, \dots, \sigma_{rn}^2, \sigma_{r1}^4, \dots, \sigma_{rn}^4\}, syn, IN, OUT)$ where

(1) The proposition neurons have the form $\sigma_{pi}^h = \{0, w_{ij}^h, \lambda_i^h, r_i^h\}$, where h and i denote the label of the layers in the network and the label of proposition neurons respectively.

(a) 0 denotes that the potential value is zero in all proposition neurons.

(b) The synapses connecting the proposition neurons and rule neurons are associated with weights $w_{ij}^1 = rand(0,1), i = 1, \dots, k, j = 1, \dots, n$ and $w_{ij}^3 = 1, j = 1, \dots, n$. The learning strategy associated with this model is to find a set of weights with the highest accuracy rate.

(c) r_i^h represents the firing/spiking rule and it is of the form $r_i^1 : E^1 / a^{\alpha_i} \rightarrow a^{\alpha_i}, i = 1, \dots, k$, where $E^1 = \{\alpha_i \geq \lambda_i^1\}$, and $\lambda_i^1 = 0, i = 1, \dots, k$. The last neuron is called as bias neuron and $\alpha_k = 1$. The firing/spiking rule present in the third layer has the form $r_1^3 : E^3 / a^p \rightarrow a^p$, where $E^3 = \{p \geq \lambda_1^3\}, \lambda_1^3 = 0$ and $p = \theta_1 \vee \theta_2 \vee \dots \vee \theta_n$. The firing rule present in the last layer is of the form $r_1^5 : a \rightarrow a$.

(2) The rule neurons are of the form $\sigma_{rj}^k = \{0, 1, \tau_j^k, r_j^k\}$, where k and j denote the label of the layers in the network and the label of rule neurons respectively.

(a) The potential value of all rule neurons is 0.

(b) The confidence factor/truth value of all rule neurons is 1.

(c) The weight 1 is associated with the synapses connecting the rule neurons and proposition neurons, since these weights are not adjusted using the weight adjustment process.

(d) The rule neurons $\sigma_{r1}^2, \dots, \sigma_{rn}^2$ have spiking rules of the form $r_j^2 : E^2 / a^{\theta_j} \rightarrow a^{\theta_j}, j = 1, 2, \dots, n$, where $E^2 = \{\theta_j \geq \tau_j^2\}, \tau_j^2 = 0, \theta_j = (w_{1j} \otimes \alpha_1) \oplus (w_{2j} \otimes \alpha_2) \oplus \dots \oplus (w_{kj} \otimes \alpha_k)$. The multiplication operator and addition operator of fuzzy truth values are denoted by \otimes and

\oplus respectively. Furthermore, the rule neurons $\sigma_{r1}^4, \dots, \sigma_{rn}^4$ have the spiking rules of the form $r_1^4 : E_j^4 / a^{\theta_j} \rightarrow a; d_j$, where $E_j^4 = \{\theta_j \geq \tau_j^4\}$ and $\tau_j^4 = p$.

(3) $syn = \{(\sigma_{p1}^1, \sigma_{r2}^2), (\sigma_{p1}^1, \sigma_{r2}^2), \dots, (\sigma_{pk}^1, \sigma_{rn}^2), (\sigma_{r1}^2, \sigma_{p1}^3), (\sigma_{r2}^2, \sigma_{p1}^3), \dots, (\sigma_{rn}^2, \sigma_{p1}^3), (\sigma_{p1}^3, \sigma_{r1}^4), (\sigma_{p1}^3, \sigma_{r2}^4), \dots, (\sigma_{p1}^3, \sigma_{rn}^4), (\sigma_{r1}^4, \sigma_{p1}^5), (\sigma_{r2}^4, \sigma_{p1}^5), \dots, (\sigma_{rn}^4, \sigma_{p1}^5)\}$.

(4) $IN = \{\sigma_{p1}^1, \dots, \sigma_{pk}^1\}, OUT = \{\sigma_{p1}^5\}$.

In large power transformers, the insulation and heat dissipation are done by transformer oil. Under different thermal and electricity condition, the oil present in the transformer decomposes. If any faults occurs in the oil-immersed power transformers, by analyzing the transformer oil the faults can be identified efficiently. Furthermore, while performing this task the ratios between the gases are given as inputs to the network. In the LSNPS model constructed in [91], the ratio of CH_4/H_2 , C_2H_4/C_2H_6 , C_2H_2/C_2H_4 , $C_2H_2/Total Hydrocarbon$, $C_2H_4/Total Hydrocarbon$, $CH_4/Total Hydrocarbon$, $(C_2H_4 + CH_4)/Total Hydrocarbon$ are considered as input of the system. In this model, the weights associated with the synapses are trained using the Widrow-Hoff algorithm.

The LSNPS model can express the relationship between the DGA data in transformer oil and different types of faults in transformers efficiently. Moreover, LSNPS model with belief AdaBoost is very efficient while diagnosing faults in transformers for thermal and electric fault situations with dissolved gas data (DGA data). Another purpose of using the belief AdaBoost algorithm is to improve the generalization ability. While performing the task of fault diagnosis, the belief AdaBoost is able to avoid the disadvantages of classical AdaBoost algorithm. Moreover, the accuracy of the test set is improved. Altogether, belief AdaBoost improves the generalization ability of learning systems. Finally, if the correctness of diagnosis results are considered, the proposed method in [91] is superior to well-known methods such as improved three-ratio method, back-propagation neural network, support vector machine, deep belief network.

2.2. Power Transmission Networks

This section discusses the fault diagnosis methods for power transmission networks. Moreover, the fault diagnosis methods discussed in this section are based on three different variants of spiking neural P systems, i.e., FRSNPS with trapezoidal fuzzy numbers (FD-SNPS), SNPS with self-updating rules (srSNPS) and interval-valued fuzzy spiking neural P system (IVFSNPS).

The main components of power transmission networks are mainly divided into two types, i.e., transmission equipment and converting equipment. These equipment also contain many components. Power transmission networks are one of the major networks in the power systems. So the investigation of the fault diagnosis in power transmission network is an important topic. The study in [89] investigated this topic using the status of the protective relays and circuit breakers received from the power SCADA systems. More specifically, the faults occurring in the lines, buses and transformers are investigated.

In [89], the task of fault diagnosis of power transmission networks is performed by using the FRSNPS with trapezoidal fuzzy numbers, i.e., FDSNPS. The use of trapezoidal fuzzy numbers makes these models suitable for practical applications. Moreover, using trapezoidal fuzzy numbers seems to be beneficial while handling the uncertainties associated with the approaches of practical applications. The FDSNPS model has been used to perform the task of fault diagnosis of main sections, transmission lines, buses and transformers, in power transmission networks. Note that the task of fault diagnosis consists of five steps: estimating outage areas, identifying candidate faulty sections, building a fault diagnosis model for each candidate section in each outage area, initializing inputs for each diagnosis model, each model performing its reasoning algorithm and determining faulty sections with fault confidence levels. At first incomplete and uncertain status information which are known as operation messages are received from SCADA system. Next, the outage areas are estimated using the network topology analysis method to obtain the faulty sections. Then the fault diagnosis model is constructed for each candidate section in

the outage area and they are constructed based on the static data, network topology and protection configuration of a power system. This is followed by performing of the fuzzy reasoning algorithm which obtains confidence levels of suspicious fault sections. At the end, according to the fault confidence level, the fault sections are identified.

Now we discuss the fault diagnosis process in detail and in order to do so at first the structure and working of FDSNPS model are discussed in detail [89].

The FRSNPS with trapezoidal fuzzy numbers (FDSNPS) have a similar structure like FRSNPS models. An FRSN P system with trapezoidal fuzzy numbers (with degree $m \geq 1$) is a $(m + 4)$ -tuple $\Pi = (O, \sigma_1, \dots, \sigma_m, syn, IN, OUT)$. However, unlike in the case of FRSNPS, the neurons $\sigma_1, \dots, \sigma_m$ are of the form $\sigma_i = (\theta_i, c_i, r_i), 1 \leq i \leq m$ where (a) $\theta_i \in [0, 1]$ is a trapezoidal fuzzy number and it represents the potential value of spikes (i.e., the value of electrical impulses) contained in neuron σ_i ; (b) $c_i \in [0, 1]$ is a trapezoidal fuzzy number and it represents the fuzzy truth value of the neuron σ_i ; (c) r_i represents the firing(spiking) rule present in the neuron σ_i . These rules are of the form $E/a^\theta \rightarrow a^\beta$ where E is a regular expression. Moreover, θ and β are trapezoidal fuzzy numbers in $[0, 1]$.

The fuzzy production rules of Type-1, Type-2, Type-3 and Type-4 are modeled using FRSNPS with trapezoidal fuzzy numbers. These types of rules are as follows:

Type 1: $R_i(c_i) : p_j(\theta_j) \rightarrow p_k(\theta_k) (1 \leq j, k \leq N_p)$.

Type 2: $R_i(c_i) : p_1(\theta_1) \otimes \dots \otimes p_{k_1}(\theta_{k_1}) \rightarrow p_k(\theta_k)$.

Type 3: $R_i(c_i) : p_1(\theta_1) \rightarrow p_2(\theta_2) \otimes \dots \otimes p_k(\theta_k)$.

Type 4: $R_i(c_i) : p_1(\theta_1) \otimes \dots \otimes p_{k_1}(\theta_{k_1}) \rightarrow p_k(\theta_k)$.

Moreover, modeling of the above rules requires two types of neurons, i.e., proposition neurons and rule neurons. The rule neurons of the FDSNPS model consist of three types: *general*, *and* and *or*. Details can be found in [89].

The reasoning algorithm introduced in [89] is based on FDSNPS. The main components of the reasoning algorithm are neuron's firing mechanism and matrix operations. In [89], the truth values associated with proposition neurons are trapezoidal fuzzy numbers. Next, by using the reasoning algorithms, fuzzy truth values of the proposition neurons with unknown truth values are obtained. Moreover, the truth values of these neurons can be derived after the inputs are received in some proposition neurons.

The proposed fault diagnosis method in [89] has good fault tolerant capacity. It also can efficiently handle incomplete and uncertain messages received from SCADA systems. Since the relationship between the faults, CBs and PRs can be represented graphically and using simple mathematical expressions, the proposed fault diagnosis model is easily understandable. Moreover, trapezoidal fuzzy numbers can express these relationships more precisely. Altogether, this method has good accuracy and in some situations is superior than the traditional methods such as fuzzy logic (FL), fuzzy Petri nets (FPN) and genetic algorithm (GA).

In [125], another method of fault diagnosis for power transmission networks is recently introduced. This fault diagnosis method is based on a new class of SNPS, i.e., SNPS with self-updating rules (srSNPS). This model is inspired by the apoptosis mechanism in biology and reduction ability of the rough sets. Moreover, these features are incorporated into the framework of SNPS. These features are useful and simplifies the complexities of the srSNPS model and also can process the uncertain and incomplete messages received from SCADA efficiently. The structure of spiking neural P systems with self-updating rules is as follows:

A spiking neural P system with self-updating rules (srSNPS) [125] is a tuple: $\Pi = (O, M_e, \sigma_1, \dots, \sigma_m, syn, IN, OUT)$ where $M_e = (D_i, C_j)$ represents microenvironment.

Again, $D_i = (\theta_{di/T}, T, f_i), 1 \leq i \leq s$, represents the i -th decision-making neuron (DN) in M_e . It also represents suspicious faulty equipment in the targeted power network.

(i) The value of $\theta_{di/T}$ is either equals to 0 or 1 and it denotes the pulse value of the i -th DN at time T . When the suspicious faulty equipment corresponding to D_i faults, the value of $\theta_{di/T}$ is 1 and vice versa.

(ii) T represents the sequence time of spikes, and at each unit time the decision-making neuron DN produces a spike. In this manner, a pulse value sequence is formed and it is then recorded in the DN after T unit time.

(iii) f_i represents a forgetting rule of the i -th DN and it is of the form $E/\{a^{\theta_{di}/T} \rightarrow \lambda; g = 0\}$. The g in the forgetting rule represents the reasoning step of the reasoning algorithm of the srSNPS, i.e., the self-updating matrix reasoning algorithm. After application of the rule f_i , the spikes in the D_i are emptied and the calculation process will be reinitialized (i.e., $g = 0$). The firing condition $E = \{\theta'_{di}/T \neq \theta_{di}/T\}$ represents that the pulse value in D_i has changed.

(b) $C_j = (\theta_{cj}/T, T, f_j, A_{rj}), 1 \leq j \leq t$, represents the j -th condition neuron (CN) in M_e . It also represents a protective relay or circuit breaker in the targeted power transmission network.

(i) The value of θ_{cj}/T in C_j is either equals to 0 or 1. It also represents the pulse value of the j -th CN at time T . When the protection device corresponding to C_j acts, the value of θ_{cj}/T is 1 and vice versa.

(ii) T represents the sequence time of spikes, and a spike is produced by CN at each unit time. Next, a pulse value sequence is formed and after T unit time it is recorded in the CN.

(iii) f_j represents the forgetting rule of the j -th CN and it is of the form $E/\{a^{\theta_{cj}/T} \rightarrow \lambda; g = 0\}$.

After application of the rule f_j , the spikes in C_j are emptied and the calculation process will be reinitialized (i.e., $g = 0$). The firing condition $E = \{\theta'_{cj}/T \neq \theta_{cj}/T\}$ represents that pulse value in C_j has changed.

(c) A_{rj} represents an apoptosis rule in M_e and it is of the form $E/\{C_j\} \rightarrow \{Algorithm1\{L\}\}$. When A_{rj} is applied, then apoptosis rule in C_j is executed in order to determine whether it lives or dies. It means that CNs execute "Apoptosis algorithm for condition neurons" and as output the survival condition neuron (containing important information) set L is obtained. During this period, CNs and redundant information die and they do not participate in fault reasoning any further. Note that the firing condition $E = \{g = 0 \cap sig_j = 0\}$ indicates that the rule A_{rj} can be applied if and only if in the initial configuration (i.e., $g = 0$) $sig_j = 0$.

(3) $\sigma_i = (\theta_i, r_i, e_i), 1 \leq i \leq p$, represents the i -th proposition neuron (PN) corresponding to a protection device or the equipment. Similarly, $\sigma_j = (\delta_j, r_j), 1 \leq j \leq q$, represents the j -th rule neuron (RN) corresponding to a fault production rule where $p + q = m$.

(a) θ_i and δ_j represent the pulse values (equal to 0 or 1) in proposition neuron σ_i and rule neuron σ_j , respectively.

(b) r_i represents a firing rule of σ_i with the form $E/a^\theta \rightarrow a^\theta$, where $E = \{a\}$ represents the firing condition. r_i is applicable when σ_i contains a spike. At this time, σ_i consumes a spike with pulse value θ , produce a new spike with the same pulse value, and transmits it to its postsynaptic neurons. r_j represents a firing rule of σ_j with the form $E/a^\delta \rightarrow a^\beta$, where $E = \{a\}$ is the firing condition. The firing condition represents that once σ_j contains a spike, r_j can be applied and σ_j consumes a spike with pulse value δ and subsequently produces a spike with pulse value β (equals to 0 or 1). Next, the spike is transmitted to its postsynaptic neurons.

(c) e_i represents a self-updating rule of the form $E/a^\theta \rightarrow a^\theta$. The firing condition is $E = \{\epsilon_i > 0\}$, represents that the self-updating rule e_i can be applied if and only if the self-updating operator $\epsilon_i > 0$. The neuron σ_i consumes a spike with pulse value θ and produces a new spike with pulse value $\bar{\theta}$ (equals to 0 or 1, called the antispikes of θ), and next $\epsilon_i = \epsilon_i - 1$ is performed. It is important to note that only input proposition neurons contain self-updating rules.

Next the reasoning algorithm based on srSNPS is introduced. This reasoning algorithm is called as self-updating reasoning algorithm [125]. Similarly, as in the previous methods, this fault diagnosis method also reasons out the fuzzy truth values of other unknown fuzzy propositions from known fuzzy propositions (input neurons). However, in this case, the

entries of the the truth value vectors are real numbers. Two main components of this fault diagnosis method are (i) Apoptosis algorithm for condition neurons and (ii) Self-updating matrix reasoning algorithm. These algorithms play an important role in the fault diagnosis process which is divided into five components, i.e., (1) transmission network partition, (2) the SNPS model establishment, (3) the pulse value correction and (4) computing, and (5) the protection device behavior evaluation. In order to save the diagnosis time, the first two components can be finished before any fault occurs in the system. The apoptosis algorithm proposed in [125] is based on the framework of srSNPS where the reduction function of rough sets is also incorporated into the model. These features help in processing incomplete and uncertain information received from SCADA efficiently. Moreover, during this process the models comprehensively use fault information including action information, start information, and overlimit signals of protection devices.

In the fault diagnosis method proposed in [125] along with apoptosis algorithm for condition neurons and Self-updating matrix reasoning algorithm, the depth-first search algorithm (DSA), the weight network segmentation method (WNSM) and the protection device event tree (PDET) are also used. Some of the advantages of the proposed method is that it can deal with uncertain and incomplete fault alarm messages without historical statistic and expert experience. Furthermore, the apoptosis algorithm for CNs removes the redundant fault information before modeling. These features simplifies the problem complexity and the transmission network partition improves the topological adaptive ability. From the case studies, it is also inferred that the proposed method based on srSNPS has high diagnostic accuracy and fault tolerance with good diagnosis result interpretability and fast speed. Furthermore, in comparison with cause-effect network (CEN), fuzzy Petri net (FPN) and fuzzy reasoning spiking neural P system (FRSNPS), the fault diagnosis model based on srSNPS, uses less number of neurons.

Another method of fault diagnosis of power transmission networks was introduced in [129]. The variant of SNPS used in this method is different from the variants mentioned above. It is known as interval-valued fuzzy spiking neural P system (IVFSNPS). The IVFSNPS model can efficiently describe the incomplete and uncertain messages received from SCADA systems. Moreover, the causality between the faulty sections and corresponding protective relays and circuit breakers can be modeled by this model graphically. It also can identify the faulty sections in power transmission networks accurately and effectively.

Now in order to understand the fault diagnosis process we must discuss the structure and working of IVFSNPS.

Definition 3. An IVFSNP system (IVFSNPS) [129] of degree m is a construct of the form

$\Pi = (O, \sigma_1, \sigma_2, \dots, \sigma_m, syn, IN, OUT)$ where:

(2) $\sigma_1, \sigma_2, \dots, \sigma_m$ represents the neurons and $\sigma_i = (\theta_i, c_i, r_i)$ $i \in \{1, 2, \dots, m\}$, where:

(a) θ_i is an interval-valued fuzzy number and it represents the value of spikes initially contained in neuron σ_i ;

(b) c_i is an interval-valued fuzzy number and it represents the confidence factor for a fuzzy production rule. Sometimes it is omitted while representing a fuzzy proposition;

(c) r_i is a firing rule of neuron σ_i , with the form $a^\theta \rightarrow a^\theta$ or $a^\theta \rightarrow a^\beta$, where θ and β are two interval-valued fuzzy numbers.

The fault diagnosis method of power transmission networks identify the faulty components using the information of tripping PRs and CBs received from the SCADA systems. The method in [129] considered fault diagnosis for both lines and buses. In [129], the fault diagnosis method is realized by IVFSNPS. Like in the earlier fault diagnosis method of power transmission [84,89,91], in this method IVSNPS contains three types of neurons (proposition neuron and two types of rule neurons). However, the rule neurons are of different types, i.e., $\textcircled{\Delta}$ -type rule neurons and $\textcircled{\nabla}$ -type rule neurons. IVSNPS model helps us to understand the relationship between the faulty sections, PRs and CBs. Next, the fuzzy reasoning algorithm is performed in order to obtain the degree of confidence of

the faulty sections. It is done by reasoning out truth values of proposition neurons. In this algorithm, synaptic connections between the proposition neurons and rule neurons are represented using a matrix and it is further used to reason out the truth values of the proposition neurons without any truth value associated with them. It is important to note that in the fault diagnosis method based on IVFSNPS, the truth values associated with the proposition neurons are integer valued fuzzy numbers.

Any large scale power transmission network has many sections. In case of identification of faults in complex and large system, for each system a corresponding IVFSNPS subsystem is constructed. Furthermore, the fault diagnosis model for large scale power transmission network consists of many IVFSNPS subsystems. These systems work in parallel and it makes the fault diagnosis process much faster. This fault diagnosis method also can determine incorrect signals. So this method is suitable for identification of faults in complex power transmission networks. It also has good fault tolerance capacity.

2.3. Traction Power Supply Systems

This section discusses the use of another variant of FRSNPS, i.e., weighted fuzzy reasoning spiking neural P systems (WFRSNPS) for performing the task of fault diagnosis in traction power supply systems of high-speed railways.

A new method to perform the task of fault diagnosis in traction power supply system of high-speed railways (TPSSs) was introduced in 2015 [88]. Moreover, a new variant of FRSNPS, i.e., WFRSNPS (weighted fuzzy reasoning spiking neural P system) was used to perform this task. These models can efficiently express the status information of the protective relays and circuit breakers received from the SCADA system and then perform weighted matrix-based fuzzy reasoning algorithm to obtain fault confidence levels of the faulty sections.

Problem description: Whenever any fault occurs in the power systems, at first the fault section is identified by the protective relays (PRs) and then the corresponding circuit breakers (CBs) are tripped in order to isolate the fault. It ensures that the other parts of the power system work normally. The main purpose of any fault diagnosis model is to identify the faults by analyzing the protective relay and circuit breaker information received from the SCADA system. In the proposed method in [88], a fault diagnosis unit is represented by a feeding section fed by a same traction substation. It is possible because the feeding sections work independently. Furthermore, after receiving the status information of PRs and CBs from SCADA, the faults on feeder lines, buses, traction transformers and auto-transformers in TSSs (traction substations), ATPs and section posts (SPs) are identified.

Now we explain how the fault diagnosis for traction power supply system of high-speed railways (TPSSs) is performed using the framework of WFRSNPS. So, at first the structure and functioning of weighted fuzzy spiking neural P systems need to be discussed.

Definition 4. A WFRSN P system (WFRSNPS) [88] of degree $m \geq 1$ is a $(m + 4)$ -tuple $\Pi = (O, \sigma_1, \dots, \sigma_m, \text{syn}, \text{IN}, \text{OUT})$, where:

- (1) neurons $\sigma_1, \dots, \sigma_m$ are of the form $\sigma_i = (\theta_i, c_i, \vec{\omega}_i, \lambda_i, r_i), 1 \leq i \leq m$ where:
 - (a) $\theta_i \in [0, 1]$ is a real number which represents the potential value/electric impulses of spikes in neuron σ_i ;
 - (b) $c_i \in [0, 1]$ is a real number which represents the truth value associated with neuron σ_i ;
 - (c) $\vec{\omega}_i = (\omega_{i1}, \dots, \omega_{iN_i})$ is a vector where each element of the tuple is a real number in $(0, 1]$ and $\omega_{ij} (1 \leq j \leq N_i)$ represents the weight which is present on the j th output synapse of neuron σ_i . Moreover, $N_i \in \mathbb{N}$ and it represents the number of synapses from neuron σ_i . The vector $\vec{\omega}_i$ represents the output weight vector of neuron σ_i .
 - (d) The firing threshold of neuron σ_i is denoted by the real number $\lambda_i \in [0, 1)$;
 - (e) r_i represents a firing /spiking rule present in σ_i . These rules are of the form $E / a^\theta \rightarrow a^\beta$, where $\theta, \beta \in [0, 1]$ are real numbers. The threshold/firing condition is different from the FRSNPS models [84] and it is a combination of two conditions, i.e., $E = \{a^n, \theta \geq \lambda_i\}$. So the rule is

applicable if and only if neuron σ_i receives at least n spikes from the antecedent neurons and the potential value satisfies the condition $\theta \geq \lambda_i$;

Similarly, as in [84,89], the neurons in WFRSNPS are also composed of two types, i.e., proposition and rule neurons. The rule neurons consist of three types, i.e., *general*, *and*, and *or*. The proposition neurons work as input neurons. The working of the *general*, *and*, and *or* rule neurons is similar to working of the rules neurons in FDSNPS [89]. However, it has the following distinct features:

(1) The *general* rule neuron is denoted by $R_{(c,general)}$ and after application of the firing rule, the newly obtained spike has the potential value $\beta = \theta * \omega * c$.

(2) The *and* rule neuron is represented by $R_{(c,and)}$ and the newly obtained spike has the potential value $\beta = [(\theta_1 * \omega_1 + \dots + \theta_k * \omega_k) / (\omega_1 + \dots + \omega_k)] * c$, where the neuron is connected with k antecedent neurons.

(3) Similarly, $R_{(c,or)}$ represents the *or* rule neuron and the potential value of the newly obtained spike is $\beta = \max\{\theta_1 * \omega_1, \dots, \theta_k * \omega_k\} * c$.

Next the fuzzy production rules are modeled in the framework of WFRSNPS models. The fuzzy production rules/ fault diagnosis production rules in [88] are divided into four types. These rules have the following forms:

Type 1 (Simple Rules) R_i : IF $p_j(\theta_j)$ THEN $p_k(\theta_k)(CF = c_i)$,

Type 2 (Compound *and* Rules) R_i : IF $p_1(\theta_1)$ and ... and $p_{k-1}(\theta_{k-1})$ THEN $p_k(\theta_k)(CF = c_i)$,

Type 3 (Compound *or* Rules) R_i : IF $p_1(\theta_1)$ or ... $p_{k-1}(\theta_{k-1})$ THEN $p_k(\theta_k)(CF = c_i)$,

Type 4 (Conditional *and* Rules) R_i : WHEN $p_0(\theta_0)$ is true, IF $p_1(\theta_1)$ and ... and $p_{k-1}(\theta_k)$ THEN $p_k(\theta_k)(CF = c_i)$.

The corresponding fuzzy production rules in the framework of WFRSNPS models can be referred to [88].

WFRSNPS model is similar to FRSNPS with one exception. In case of FRSNPS, no weight was associated with the synapses connecting the neurons. However, in WFRSNPS, weights are associated with the synapses connecting the neurons. Moreover, the proposed model can express status information of protective relays and circuit breakers efficiently. Next, the reasoning algorithm based on WFRSNPS is performed in order to identify the faults. The reasoning algorithm is called as weighted matrix-based reasoning algorithm (WMBRA). The synaptic connections are also represented by weighted matrix. Unlike in [89], the entries of the truth value vectors are real numbers and using the values received by the input proposition neurons and the truth values of these neurons, the truth values of other proposition neurons without any truth values associated with them are obtained.

2.4. Metro Traction Power Systems

The framework for solving fault diagnosis of metro traction power systems was introduced in [82]. This task of fault diagnosis was performed by a modified variant of FRSNPS, i.e., MFRSNPS. In [82], the proposition and rule neurons are introduced to represent the information about the protective relays and circuit breakers from SCADA. Then the fuzzy reasoning algorithm based on the MFRSNPS is performed to identify the fault section in the metro traction power supply systems.

Problem description: Metros are one of the most important mode of urban transport in 21st century. Metro traction power supply systems (MTPSSs) are considered as a special variant of distribution networks with direct current (DC) power supply. Traction power supply systems are also known as the energy systems of rail transportation systems. It also ensures the safe and reliable operations in trains. The operations of MTPSSs can be impacted by failure of equipments, interruption of power services etc. So in order to keep the working of MTPSSs smooth, it is necessary to design fault diagnosis methods where the fault sections can be identified by the dispatchers efficiently and the power supply to the system is restored as soon as possible. The fault diagnosis of MTPSS is considered very challenging because of mainly two reasons, i.e., (1) two-way feeding power supply approach; (2) complex protection system caused by the DC power supply mode. In [82], a

fault diagnosis approach for MTPSS was introduced based on modified FRSNPS. However, the framework of the fault diagnosis method is similar to the framework as mentioned in [89].

MFRSNPS is another variant of FRSNPS and the structure of this model is similar to FRSNPS with some exceptions. The structure and working of the MFRSNPS model is as follows.

Definition 5. A MFRSN P system (MFRSNPS) [82] of degree $m \geq 1$ is a $(m + 4)$ -tuple $\Pi = (O, \sigma_1, \dots, \sigma_m, \text{syn}, \text{IN}, \text{OUT})$, where:

- (1) The neurons $\sigma_1, \dots, \sigma_m$ are of the form $\sigma_i = (\theta_i, c_i, r_i, \lambda_i)$, $1 \leq i \leq m$, and
 - (a) $\theta_i \in [0, 1]$ is a real number which represents the potential value of spikes in neuron σ_i ;
 - (b) $c_i \in [0, 1]$ is a real number which represents the truth value of neuron σ_i ;
 - (c) r_i represents a firing/spiking rule in neuron σ_i and it is of the form $E/a^\theta \rightarrow a^\beta$, where $\theta, \beta \in [0, 1]$ are real numbers. The firing condition is $E = \{a^\theta, \theta > \lambda_i\}$ and it means that the rule is applicable if and only if neuron σ_i receives at least n spikes as input or from the antecedent neurons and $\theta > \lambda_i$ ($\lambda_i \in [0, 1]$ is a real number which represents the firing threshold of neuron σ_i). Otherwise, it is not applicable. Furthermore, after application of this rule in neuron σ_i , a spike with potential value θ is consumed and a spike with potential value β is produced.
 - (d) λ_i is a real number in $[0, 1]$ representing the firing threshold of neuron σ_i ;

The fault diagnosis rules considered for MTPSSs are different from the rules in [84,88,89,129] and they are of three types:

- (1) General Rule R_i : IF $p_j(\theta_j)$ THEN $p_k(\theta_k)(CF = c_i)$, where p_j and p_k are propositions, $c_i \in [0, 1]$ is a real number (certainty factor of such rule R_i); $\theta_j, \theta_k \in [0, 1]$ are real numbers (truth values of p_j and p_k); $\theta_k = \theta_j * c_i$ (truth value of p_k).
- (2) And Rule R_i : IF $p_1(\theta_1)$ and ...and $p_{k-1}(\theta_{k-1})$ THEN $p_k(\theta_k)(CF = c_i)$, where p_1, \dots, p_k (propositions); $c_i \in [0, 1]$ is a real number (certainty factor of R_i); $\theta_1, \dots, \theta_k \in [0, 1]$ are real numbers (truth values of p_1, \dots, p_k); $\theta_k = \frac{[(\theta_1 + \dots + \theta_{k-1})]}{(k-1)} * c_i$ (truth value of p_k).
- (3) Or Rule R_i : IF $p_1(\theta_1)$ or ...or $p_{k-1}(\theta_{k-1})$ THEN $p_k(\theta_k)(CF = c_i)$ p_1, \dots, p_k (propositions); $c_i \in [0, 1]$ is a real number (certainty factor of R_i); $\theta_1, \dots, \theta_k \in [0, 1]$ are real numbers (the truth values of p_1, \dots, p_k); $\theta_k = \max\{\theta_1 * c_i, \dots, \theta_{k-1} * c_i\}$ (Truth value of p_k).

These rules are reconstructed in the framework of MFRSNPS. Again similarly as in [84,88,89,129], there exist some neurons without any truth value associated with them. The truth values of these neurons are obtained by reasoning algorithm. The reasoning algorithm introduced in [82] is based on MFRSNPS. Moreover, the entries of the truth value vectors are real numbers. Furthermore, the synaptic connections between the neurons are represented by matrices and there exist some proposition neurons which work as input neurons. Next, using these values, the truth values of the above mentioned proposition neurons are obtained.

The fault diagnosis method based on MFRSNPS has some advantages. It can represent the operation information of PRs and CBs efficiently. Moreover, it can accurately identify faults with certain/uncertain and complete/incomplete operate information.

2.5. Fault Section Estimation of Power Systems

This section discusses the use of a novel model in SNPS, i.e., optimization spiking neural P systems (OSNPS) [130], for fault section estimation of power systems. OSNPS was introduced in 2014 with the purpose of solving optimization problems using the framework of spiking neural P system. In [52,126], a framework based on OSNPS was introduced to solve the power system fault section estimation (FSE) problem. FSE can be formulated as an optimization problem. OSNPS models are very efficient and after receiving the PRs and CBs information from SCADA system, these models are capable of searching and output fault sections automatically. Furthermore, these models are very efficient while identifying single as well as multiple faults even with incomplete and uncertain messages received from the SCADA system.

Problem Description: The fundamental purpose of FSE problem in power system is to obtain a fault hypothesis which is capable of explaining warning signals with maximum degree of confidence and it is based on optimization methods. It is also possible to formulate the FSE problem as 0 – 1 programming problem with an objective function as in the Equation (5). This equation is derived from the causality between a fault and the statuses of PRs and CBs received from the SCADA system. The fault hypothesis is obtained by using the optimization method. More specifically, by finding the minimum value of

$$E(S) = \sum_{j=1}^{n_c} |c_j - c_j^*(S, R)| + \sum_{k=1}^{n_r} |r_k - r_k^*(S)| \tag{5}$$

where:

- (1) n_c represents the numbers of circuit breakers (CBs) and n_r represents the number of protective relays;
- (2) $E(S)$ is status function of all the sections in a power system;
- (3) The status of sections in a power system is represented by an n -vector S . The number n represents the number of sections: when section i is faulty, then $s_i = 1$; otherwise, $s_i = 0, i = 1, \dots, n$;
- (4) In a protection system, the real status of the j th circuit breaker is denoted by $c_j (1 \leq j \leq n_c)$ and it also can be expressed as the j th element of an n_c -vector. When CB_j trips, then $c_j = 1$; otherwise, $c_j = 0$;
- (5) In a protection system, expected status of the j th circuit breaker is represented by $c_j^*(S, R) (1 \leq j \leq n_c)$ and it is also expressed as the j th element of an n_c -vector. When CB_j trips, then $c_j^* = 1$; otherwise, $c_j^* = 0$;
- (6) In a protection system, the real status of the k th protective relay is denoted by $r_k (1 \leq k \leq n_r)$ and it is also expressed as the k th element of an n_r -vector. When the k th protective relay is operated, then $r_k = 1$; otherwise, $r_k = 0$;
- (7) In a protection system, the expected status of the k th protective relay is denoted by $r_k^*(S) (1 \leq k \leq n_r)$ and it also can be expressed as k th element of an n_r -vector. When the k th protective relay operates, $r_k^* = 1$; otherwise, $r_k^* = 0$.

In [126], the OSNPS model has been used to minimize $E(S)$ in Equation (5) and it also helps in fulfilling the task of fault section estimation in power systems. From SCADA system the real status of protective relays and CBs can be obtained. However, from the operational principles and protection structures of power systems, the expected status of PRs and CBs can be achieved. Then the OSNPS is used to obtain the minimum value of $E(S)$ and it is done only when all the expected status and real status of PRs and CBs are obtained. Furthermore, the vector elements of S are obtained according to the minimum value of $E(S)$ in Equation (5).

Now we explain the fault diagnosis method and in order to do so at first the structure and working of the OSNPS model are described.

Definition 6. An extended spiking neural P system (ESNPS) [130] of degree $m \geq 1$, is a $(m + 5)$ -tuple $\Pi = (O, \sigma_1, \dots, \sigma_{m+2}, syn, I_0)$, where:

- (1) The neurons $\sigma_i, 1 \leq i \leq m$, are of the form $\sigma_i = (1, R_i, P_i)$, where $R_i = \{r_i^1, r_i^2\}, r_i^1 = \{a \rightarrow a\}, r_i^2 = \{a \rightarrow \lambda\}$. $P_i = \{p_i^1, p_i^2\}$ is a finite set of probabilities and the probability p_i^j is associated with rule r_i^j for $1 \leq j \leq 2$ such that $p_i^1 + p_i^2 = 1$;
- (2) $\sigma_{m+1} = \sigma_{m+2} = (1, \{a \rightarrow a\})$;
- (3) $syn = \{(i, j) | (i = m + 2 \wedge 1 \leq j \leq m + 1) \vee (i = m + 1 \wedge j = m + 2)\}$;
- (4) $I_0 = \{\sigma_1, \dots, \sigma_m\}$ is a finite set of output neurons, and the output of the system is denoted by a spike train which is formed by concatenating the outputs of the neurons $\sigma_1, \dots, \sigma_m$.

The neurons σ_{m+1} and σ_{m+2} work as a supplier of spikes to the neurons $\sigma_1, \sigma_2, \dots, \sigma_m$. Moreover, after firing at each step, the neurons get reloaded by receiving one spike from each other and the process continues. The neurons $\sigma_1, \sigma_2, \dots, \sigma_m$ contain one firing rule

with probability p_i^1 and one forgetting rule with probability p_i^2 . When the first rule is applied, the output of the i th neuron is 1 with a probability p_i^1 associated with it. When the second rule is applied, the output of the neuron is 0 with a probability p_i^2 . So a spike train of 0's and 1's is obtained at any given time. Furthermore, the output of the system can be controlled by adjusting the probabilities p_1^1, \dots, p_m^1 .

OSNPS is a combination of family of ESNPS where a guider is associated with each rule inside each of the neurons in ENPS to adjust the selection probabilities of the rules. The OSNPS is a combination of H ESNPS, i.e., $ESNPS_1, \dots, ESNPS_H$ and each of these ESNPS has the structure mentioned in [130]. A guider algorithm is also added with the OSNPS models to adjust the probabilities associated with the rules present in the neurons. The pseudocode of the guider algorithm can be referred to [130].

Furthermore, the process of solving FSE problem and identifying the fault sections using the OSNPS model can be explained by the sketch shown in Figure 1:

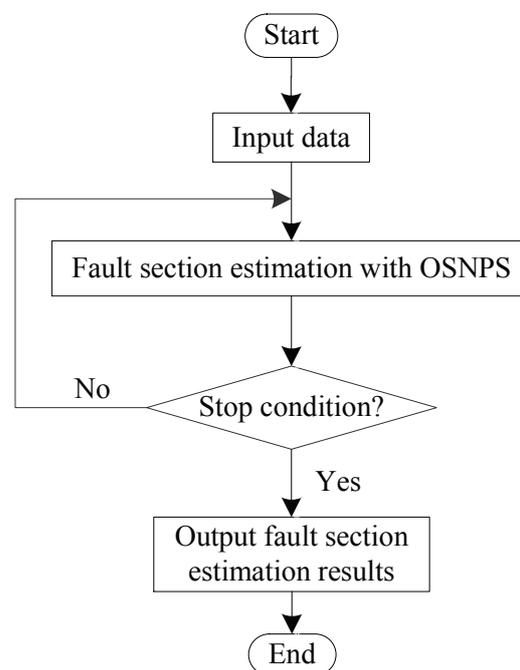


Figure 1. The sketch map of fault section estimation based on OSNPS.

Now in order to find the fault section, a fault section estimation algorithm (FSE algorithm) based on OSNPS is introduced. FSE algorithm is different from the reasoning algorithms discussed in [82,84,88,89,125,129]. These reasoning algorithms are used for obtaining the truth value of the proposition neurons which do not have any truth value associated with them. In FSE algorithm, m (number of neurons in every ESNPS), M (iteration), $p_j^a, 1 \leq j \leq m$ (learning probabilities), Δ (learning rate) and H (numbers of ESNPS) are given as input and a global optimal value G_{best} (a m -vector representing the status of the sections) is obtained as an output. Furthermore, if $G_{besti} = 1$, then it is considered that the i th section is faulty. In the reasoning algorithms in [82,84,88,89,125,129], matrices are used to represent the synaptic connections between the neurons. In FSE algorithm, instead of the connection matrices, fitness function ($E(S)$), probability matrix (P_R) are used to obtain the required output.

2.6. Fault Location of Distribution Network

Many new types of small, grid-connected or distribution system-connected devices can perform electric generation and storage. It is known as “Distribution generation”. In [86], a fault location identification method was introduced for distribution networks with distributed generation using improved spiking neural P system with anti-spikes

(IASNPS). In this method, at first the distribution networks are modeled using IASNPS and then the faults are located using the matrix algorithm.

Problem Description: The electric generation and storage performed by small, grid-connected or DER (distributed energy resources) are called as DG (distributed generation). It also works as an emerging and small power supply connected to 35 kV and below distribution networks. It is important to note that the structure of distribution networks can be transformed into multiple power supply from the single power supply when DG is connected with the distribution networks. So whenever any line fails, the type, position and capacity of the DG affects the direction and magnitude of the short circuit current. Furthermore, the impact of the DG connected to the distribution networks cannot be ignored when the permeability of DG and stiffness ration are greater than 10% and 20% respectively. Moreover, in this case the current protections are not applicable. So the identification of the faults accurately as well as understanding the protection in the distribution networks with DG is of great importance. In [86], a novel fault identification method in distribution networks with DG was introduced which is based on improved spiking neural P system with anti-spikes (IASNPS). These models can efficiently locate faults in distribution networks with DG when the messages received from SCADA are incomplete and uncertain. Furthermore, in the fault identification method, at first the complex distribution networks with DG are properly partitioned. Subsequently fault current information of CBs is read from the SCADA system followed by construction of corresponding fault location model for the suspected fault region. Next the faults are located using reasoning algorithm and at the end, using fault judgment condition these faults are located accurately.

Now we explain the fault identification process in detail. So we start with discussing the structure and working of IASNPS model. Spiking neural P systems with anti-spikes are fundamentally different from SNPS models. In this model, along with an alphabet for spike, there exists an alphabet symbol which is associated to represent anti-spikes. This model can be mathematically formulated in the following manner.

Definition 7. An IASNP system [86] of degree m ($m \geq 1$) is a $(m + 4)$ -tuple $\Pi = (O, \sigma_1, \dots, \sigma_m, \text{syn}, IN, OUT)$ where

- (1) $O = \{a, \bar{a}\}$ is the alphabet (a and \bar{a} are called spike and anti-spike, respectively);
- (2) The neurons $\sigma_1, \sigma_2, \dots, \sigma_m$ are of the form $\sigma_i = (\theta_i, r_i)$, $1 \leq i \leq m$, where
 - (i) $\theta_i \in \{1, -1, 0\}$ represents the potential value of the spike present in the neuron σ_i ;
 - (ii) r_i represents the rules present in the neuron σ_i and the rules are of the following form:
 - (a) $E/b^\theta \rightarrow b'^\beta, k$, where E is a regular expression over $\{a\}$ or $\{\bar{a}\}$. When the neuron σ_i have only the spike a then $\theta_i \in \{1, 0\}$ and similarly when \bar{a} is contained in σ_i , then $\theta_i \in \{1, 0\}$. Again, $k \in \{k_1, \bar{k}_1\}$. When neuron σ_i sends pulses to neurons σ_j along the synapse, it is indicated by k_1 and similarly when the neuron σ_j sends pulses to neurons σ_i along the synapse, it is indicated by \bar{k}_1 . Otherwise, $k = \emptyset$;
 - (b) $a\bar{a} \rightarrow \lambda$ is called as the annihilating rule and it has priority over spiking and forgetting rules;

In the SNP models discussed in [82,84,88,89,91,126], the neurons are divided into proposition and rule neurons. However, the neurons in IASNPS are divided into *node neurons* and *region neurons*. The node neurons are also divided into two types, i.e., node neurons with positive spikes and node neurons with anti-spikes. Moreover, note that in the case of region neuron, the neurons fire only if the number of spikes received by the region neuron is equal to the number of pre-synapses and after firing a spike with potential value θ is produced.

The main steps of the reasoning algorithm based on IASNP systems for distribution networks with DG include: (1) the fuzzy truth value of node neuron are given as input and using the reasoning algorithm; and (2) fuzzy truth values of unknown neurons are derived. More details can be referred to [86].

A new method for identification of faults in distribution network based on electrical synaptic transmission-based SNPS [67] was introduced in 2019. This variant of SNPS has some novel features such as (1) new types of synapses; (2) bidirectional model; (3) two types of neurons (4) the delay associated with the axons has been removed. In this model, the bidirectional characteristics of electric synaptic transmission is incorporated into SNPS. The bidirectional characteristic of electrical synaptic transmission is an important feature of distribution networks with DGs. Moreover, the concepts from the SNPS with rules on synapse have been used in [67]. More specifically, in the proposed model, the rules are on the electrical synapse. It infers that different rules can be used by each synapse. In the fault diagnosis model, at first fault location model based on the electrical synaptic transmission-based spiking neural P system is constructed for subareas of the distributed power distribution network. Next the reasoning algorithm is performed in order to locate the fault and verify the fault current information.

In order to explain the fault diagnosis process at first we explain the structure of the electrical synaptic transmission-based SNPS [67].

Definition 8. An electrical synaptic transmission-based spiking neural P system (ESTSNPS) [67] of degree $m \geq 1$, is an $(m + 4)$ -tuple $\Pi = (O, \sigma_1, \sigma_2, \dots, \sigma_m, \text{syn}, \text{IN/OUT}, \text{OUT/IN})$ where:

- $\sigma_1, \sigma_2, \dots, \sigma_m$, represents the neurons and $\sigma_i = (\alpha_i)$ ($1 \leq i \leq m$). Again, $\alpha_i \in (-1, 0, 1)$ and it represents the spike value;
- syn is the collection of electrical synapses between neurons and $\text{syn} \subseteq ((i, j), R(i, j))$, where
 - (1) (i, j) ($1 \leq i, j \leq m$) represents electrical synapses $i \neq j$ connecting neurons σ_i and σ_j .
 - (2) $R(i, j)$ represents a finite set of rules on electrical synapses. The rules in $R(i, j)$ are of the following form:
 - A. $E/a^\alpha \rightarrow a^\beta$ represents the firing rule; $E/a^\beta \rightarrow a^\alpha$ represents backward firing rule; where E is regular expression over $\{a\}$, $\alpha \in (-1, 0, 1)$, $\beta \in (-1, 0, 1)$.
 - B. The forgetting rule is of the form $E/a^\alpha \rightarrow \lambda$ (i.e., $\beta = 0$).
 If $\alpha = 0$ in backward firing rule, then $E/a^\beta \rightarrow \lambda$, and it is called as backward forgetting rule.
- IN/OUT and OUT/IN are input/output neuron set and output/input neuron set, respectively.

Electric synaptic transmission-based spiking neural P system is different from the traditional SNPS models. In this model, the value of the spikes in a neuron is not represented by the number of spikes or real numbers. Instead, it is represented by a discrete value from $\{-1, 0, 1\}$. The firing rules in the proposed model in [67] are also different and along with traditional firing and forgetting rule, there exist backward firing rule and backward forgetting rule which are associated with backward reasoning process. Moreover, the neurons are divided into two types, i.e., input/output and output/input neurons.

Next we explain the reasoning process for the proposed fault location identification method in distribution network with distributed generations. Since the electrical synaptic transmission-based spiking neural P system is a bidirectional system and it performs bidirectional reasoning. So the reasoning process in [67] is divided into forward reasoning process and the backward verification process. Moreover, note that results contained in the neuron are different during the forward reasoning process and the backward verification process. Unlike in [82,84,88,89,91], the neurons are divided into input/output (IN/OUT) and output/input (OUT/IN) neurons where instead of truth values, the neurons contain discrete values, i.e., the values associated with the neurons are from $\{-1, 0, 1\}$. Furthermore, during the backward reasoning process, the synaptic connection between the IN/OUT and OUT/IN neurons and the connection between the section neurons and OUT/IN neurons are represented by a direction matrix. Next, the information is used to verify whether fault current information is wrong.

The proposed method in [67] is efficient and it has the following advantages: (1) it can efficiently express the relationship between the various parts of the distribution

network and the logical relationship of the graphics and also can represent the fault location process visually;

(2) strong information processing capability because of parallel-distributed architecture;

(3) bidirectional characteristics of electrical synaptic transmission which is combined with electrical quantity (direction of current) for fault location of distribution network with distributed generations (DGs);

(4) high accuracy, less computation time, the fault diagnosis model and reasoning process is simple. Another important feature of the reasoning algorithm is that it is applied according to the bidirectional power flow characteristics of a distribution network with DGs;

(5) single faults, multiple faults, and misinformation faults in the distribution network with DGs can be located accurately and quickly by the proposed method.

(6) this method is applicable to a variety of fault conditions for distribution network with DGs. Moreover, in fault location, it can verify the fault current information, improves the reliability of the fault current information and the accuracy of the fault location result.

2.7. Fault Lines Detection

In [85], a fault diagnosis method based on FRSNPS model was introduced to solve the problem of detecting fault lines in a small current grounding systems. The authors analyzed six features of current/voltage signals in a small current grounding system. Furthermore, the analysis was done by considering transient and steady components. Moreover, a fault measure is considered and it is further used to quantify that a line has faults. Next, the idea of information gain degree was introduced in order to understand the importance of six features of current/voltage signals. Then in order to reduce the features, the concept of rough set theory was applied and it was followed by construction of fault line detection models based on the framework of FRSNPS.

Problem description: The fault diagnosis approach in [85] is considered for a 110 kV/35 kV distribution network with 6 feeders. At any time a single-phase-to-ground fault can occur in the distribution network and during this time the fault can occur in any of the 6 lines, i.e., lines 1–6 or in the bus. This fault affects the safe and stable operation of the distribution network. Moreover, the faults compromise the security of the system. So it is necessary to identify the faults in the distribution networks as soon as possible.

The fault line detection method proposed in [85] for a small current grounding system are divided into four processes, i.e., (1) feature analysis, (2) fault measure calculation, (3) feature information fusion and (4) fault line detection model construction. The framework of this method consists of zero sequence current/voltage signals, transient and steady component signal features, fault measures, information gain degree, rough set theory and fault line detection model with rFRSN P systems. Furthermore, the fault line detection method is based on fuzzy reasoning spiking neural P systems with real numbers (rFRSNPS).

Now we discuss the fault diagnosis process in detail. We start with discussing the structure and functioning of rFRSNPS. rFRSNPS models have similar structure as FRSNPS models and can be represented mathematically in the following manner.

Definition 9. An rFRSN P system [87,90] of degree $m \geq 1$ is a $(m + 4)$ -tuple $\Pi = (O, \sigma_1, \dots, \sigma_m, \text{syn}, \text{IN}, \text{OUT})$, where:

(2) The neurons $\sigma_1, \dots, \sigma_m$ are of the form $\sigma_i = (\theta_i, c_i, r_i), 1 \leq i \leq m$, where:

(a) $\theta_i \in [0, 1]$ is a real number and it represents the potential value of spikes (i.e., value of electrical impulses) contained in neuron σ_i ;

(b) $c_i \in [0, 1]$ is a real number and it represents the fuzzy truth value corresponding to neuron σ_i ;

and

(c) r_i represents the firing /spiking rule in neuron σ_i and it is of the form $E / a^\theta \rightarrow a^\beta$ where E is the firing condition and $\theta, \beta \in [0, 1]$ are real numbers;

The working of the rFRSNPS and FRSNPS [84] is similar. So in this section we do not explain it in detail. Moreover, like the FRSNPS models [84], rFRSNPS models contain two types of neurons, i.e., proposition neurons and rule neurons. The fuzzy production rules in the framework of rFRSNPS can be expressed in the similar manner as shown in Section 2.1. Since we have already described the construction of these rules in detail, we skip their explanation in this section.

2.8. Electric Locomotive Systems

The rFRSNPS models are also useful in constructing fault diagnosis method in electric locomotive systems. A new method for fault diagnosis in electric locomotive systems based on fuzzy reasoning spiking neural P systems with real numbers (rFRSNPS) was introduced in 2014 [87]. Electric locomotives consist of many subsystems. In [87], the fuzzy production rules are useful in expressing the relationship among the breakdown signals and faulty sections. Then a fault diagnosis model for subsystems is constructed based on rFRSNPS according to the fuzzy production rules.

Problem description: It is well-known that electric locomotive systems are composed of several systems where the functions associated with the subsystems are different. So these systems can be represented as a hierarchical tree structure of sections and subsystems (which can be referred to [87]). Next, the relationship among breakdown signals and faulty sections present in the subsystems are abstracted using the fuzzy production rules. Then based on these, the fault diagnosis models for the subsystems are constructed. More specifically, in [87], the fault diagnosis model based on rFRSNPS is constructed based on the causality among faulty sections, faulty subsystems, SS4 (Shaoshan4) electric locomotive systems. Furthermore, fault diagnosis method consists of the fault diagnosis for SS4 electric locomotive systems and their main subsystems, i.e., main circuit systems, power supply systems, traction and braking systems. At first the the diagnosis models are constructed for the subsystems and then these models are analyzed. Next the relationships among electric locomotive systems, its subsystems and their corresponding faulty sections are studied. At the end, the fault diagnosis model for SS4 electric locomotive systems was introduced.

Now we give a summary of the fault diagnosis method of a SS4 electric locomotive systems. As mentioned above, at first a fault diagnosis model for the main circuit systems is constructed. It is followed by construction of fault diagnosis model for power supply systems. Next the same is done for the traction and braking systems. Subsequently, fault diagnosis model for electric locomotive systems is introduced. If one or more that one subsystems of electric locomotive system, i.e., main circuit systems, power supply systems, and traction and braking systems have faults, then faults occur in the electric locomotive system.

The fuzzy production rules for electric locomotive systems are as follows:

Rule 1: IF P_1 OR P_2 OR P_3 OR P_4 THEN P_{16} (CF = 0.95)

Rule 2: IF P_5 OR P_6 OR P_7 OR P_8 THEN P_{17} (CF = 0.95)

Rule3: IF P_7 OR P_8 OR P_9 OR P_{10} OR P_{11} OR P_{12} OR P_{13} OR P_{14} OR P_{15} THEN P_{18} (CF = 0.95)

Rule 4: IF P_{16} OR P_{17} OR P_{18} THEN P_{19} (CF = 0.98)

The meaning of the propositions P_1, P_2, \dots, P_{19} are shown in Table 1 [87] and CF represents the certainty factor of the rules.

Table 1. Meaning of each proposition in fuzzy production rules for electric locomotive systems.

No	Propositions	Description
1	P_1	no current in pantograph
2	P_2	traction transformer breakdown
3	P_3	pulling motor inoperation
4	P_4	contactor breakdown
5	P_5	DC750V input breakdown
6	P_6	fan inoperation
7	P_7	main traction inverter breakdown
8	P_8	DC110V breakdown
9	P_9	control source converter plate breakdown
10	P_{10}	main protective relay breakdown
11	P_{11}	traction power controller breakdown
12	P_{12}	A/D breakdown
13	P_{13}	25/5V breakdown
14	P_{14}	linear electromotor inoperation
15	P_{15}	phase current sensor inoperation
16	P_{16}	main circuit system has faults
17	P_{17}	power supply system has faults
18	P_{18}	traction and braking system has faults
19	P_{19}	electric locomotive systems has faults

Next a fault diagnosis model based on rFRSNPS is constructed based on these fuzzy production rules, i.e., $\Pi_4 = (O, \sigma_1, \dots, \sigma_{23}, syn, IN, OUT)$ where

- (1) $O = \{a\}$ is the singleton alphabet (a is called spike);
- (2) The proposition neurons associated with the propositions P_1, P_2, \dots, P_{19} are $\sigma_1, \dots, \sigma_{19}$ respectively.
- (3) The OR rule neurons associated with the fuzzy production rules R_1, \dots, R_4 are $\sigma_{20}, \dots, \sigma_{23}$ respectively.
- (4) $syn = \{(1, 20), (2, 20), (3, 20), (4, 20), (5, 21), (6, 21), (7, 21), (7, 22), (8, 21), (8, 22), (9, 22), (10, 22), (11, 22), (12, 22), (13, 22), (14, 22), (15, 22), (16, 23), (17, 23), (18, 23), (20, 16), (21, 17), (22, 18), (23, 19)\}$;
- (5) $IN = \{\sigma_1, \dots, \sigma_{15}\}, OUT = \{\sigma_{19}\}$.

So from the above constructions we can observe that the rFRSPS models can effectively construct the fault diagnosis models for SS4 electric locomotive systems and their subsystems. However, in [87], reasoning algorithm based on rFRSNPS models was not introduced and the construction of reasoning algorithm based on this model can be an interesting direction of future research.

2.9. Comparisons

This section makes a comparison of several fault diagnosis models with spiking neural P systems, from the aspects of their applications in power systems, their features, advantages and disadvantages, and reasoning algorithms. Table 2 lists fault diagnosis models with spiking neural P systems in terms of applications, features and references. Tables 3–6 provide the advantages and disadvantages of fault diagnosis models with spiking neural P systems. Tables 7–9 compares several reasoning algorithms related to fault diagnosis models with spiking neural P systems.

Table 2. Comparisons of fault diagnosis models with spiking neural P systems.

No	Electric Equipment	Models	Features	Ref.
1	Transformers	FRSNPS	Fuzzy production rules in the framework of FRSNPS; Fuzzy reasoning algorithm based on FRSNPS; Proposition neuron: <i>and</i> and <i>or</i> rule neuron IEC ratio of gases are given as input	[84]
2	Transformers	LSNPS	Learning model in the framework of SNPS with belief AdaBoost; Use LMS algorithm for training of the weights in the network; Identify faults using DGA data;	[91]
3	Power transmission networks	FDSNPS	FRSNPS with trapezoidal fuzzy numbers; Effectively analyze uncertain and incomplete messages from SCADA; Proposition neuron; <i>general</i> , <i>and</i> , <i>or</i> rule neuron; Reasoning algorithm based on FRSNPS with trapezoidal fuzzy numbers;	[89]
4	Traction power supply systems	WFRSNPS	FRSNPS with weights associated with synapses; Weighted matrix-based fuzzy reasoning algorithm; Proposition neuron; <i>simple</i> , <i>general</i> <i>and</i> , <i>or</i> rule neuron;	[88]
5	Metro traction power systems	MFRSNPS	FRSNPS with modified threshold condition $E = \{a^\theta, \theta > \lambda_i\}$; Reasoning algorithm based on MFRSNPS; Proposition neuron; <i>general</i> , <i>and</i> , <i>or</i> rule neuron;	[82]
6	Power Systems	OSNPS	Combination of extended SNPS; Capable of searching and output fault sections automatically; Effectively identify faults with uncertain and incomplete messages from SCADA; Used for FSE problem; probabilities are associated with the rules; Guider algorithm associated with OSNPS	[130]
7	Distribution network	IASNPS	SNPS with spikes and anti-spikes; Matrix fuzzy reasoning algorithm based on IASNPS; Effectively analyze uncertain and incomplete messages from SCADA; Neurons are divided into node and region neurons;	[86]
8	Small current grounding systems	rFRSNPS	FRSNPS with real numbers; Reasoning algorithm based on rFRSNPS; Proposition neurons; rule neurons; Rough set theory is used to reduce the feature of the faults;	[85]
9	Electric Locomotive Systems	rFRSNPS	FRSNPS with real numbers; Proposition neurons; rule neuron; Fault diagnosis for SS4 locomotive systems;	[87]

Table 3. Advantages and disadvantages of fault diagnosis models for transformers with spiking neural P systems.

Device	SNPS Model	Advantages and Disadvantages
1. Transformers [84]	FRSNPS	Suitable to model fuzzy production rules in a fuzzy diagnosis knowledge base and their reasoning process; Feasibility and effectiveness in fault diagnosis; Handling incomplete and uncertain information Graphical representation; No time delay; neurons are always open; Dynamic reasoning process can be realized by the firing mechanism of neurons; Proposition neurons works as input neurons; Reasoning algorithm uses firing mechanism and matrix operations; Parallel computing model; Not much experimental data for comparison. Diagnosis model based on DGA.
2. Transformers [128]	FRSNPS	Information is represented using linguistic variables, membership functions, inference rule base, Four gas ratio (IEC 60599) signature as input to FRSNPS; Diagnosis the fault with more informative and more correctly decisions. Fault types interpretation of DGA. different; Linguistic terms (L.T) different;
3. Transformers [91]	LSNPS	Uses Widrow-Hoff learning and belief AdaBoost; More accurate than Improved Three-Ratio Method, BPNN, SVM, DBN. fault occurrence in transformer can be found through the ratio of gas Better generalization ability; Gas ratios given as input

Table 4. Advantages and disadvantages of fault diagnosis models for power transmission networks with spiking neural P systems.

Device	SNPS Model	Advantages and Disadvantages
1. Power transmission network [89]	FDSNP	Good fault tolerant capacity; Handle incomplete and uncertain messages efficiently; Simple, understandable mathematical expression to represent connections between faults, PRs, CBs; Efficient in identifying single and multiple faults; Matrix based reasoning algorithm; trapezoidal fuzzy numbers express the relationships more efficiently; In some cases, superior to to FL, FPN, and GA on correctly identifying fault sections; Good accuracy;
2. Power transmission network [125]	srSNPS	SNPS model with biological apoptosis mechanism; Self-updating matrix reasoning algorithm; Effectively unitizes the attribute reduction ability of rough sets; Simple graphical fault diagnosis model; Efficiently deal with uncertain and incomplete fault information without using historical statistics and expertise; srSNPS+ Apoptosis algorithm for condition neurons+Self-updating matrix reasoning algorithm + DSA + WSNM + PDET; good fault diagnosis capacity; high-fault tolerance, good speed; General, And, Or-rule neurons; Identify single and multiple faults; Less number of neurons w.r.t CEN, FPN, and FRSNPS; Good diagnostic accuracy.
3. Power transmission network [129]	IVFSNPS	Interval-valued fuzzy logic + SNPS; Process and incomplete and uncertain messages efficiently; CFs is expressed by interval-valued fuzzy numbers; Linguistic terms are used to describe the CFs; Perform good in large scale network; fast parallel computation; Efficient fault identification; Identify incorrect signals; Suitable to deal with complex fault;

Table 5. Advantages and disadvantages of fault diagnosis models for fault location with spiking neural P systems.

Device	SNPS Model	Advantages and Disadvantages
1. Fault section estimation of power systems [126]	OSNPS	OSNPS can automatically search and output fault sections; Effective for single fault, multiple faults and multiple faults with incomplete and uncertain information.
2. Fault location distribution network [86]	IASNPS	Express causality between regions and the associated nodes efficiently; Reasoning algorithm based on IASNPS; Locate faults more accurately and quickly; Locate single and multiple faults;
3. Fault location distribution network [67]	ESTSNPS	SNPS with new synapses, bidirectional model, two types of neurons; No delay in axon. Understandable graphical feature; Process information in parallel Bidirectional characteristics of electrical synaptic transmission; High accuracy, less computation time, simple and intuitive model and reasoning; Good for large scale DGs; Strong visibility; high versatility; Backward firing rules, backward forgetting rules, Forward reasoning process; backward reasoning process; O/I and I/O neurons; Applicable to many fault condition; verify the fault current information; improve the reliability and accuracy.
4. Fault line detection [85]	rFRSNPS	Six features of current/voltage signals in a small current grounding system are analyzed; Rough set theory is used for feature reduction Good accuracy;

Table 6. Advantages and disadvantages of fault diagnosis models for other types of power systems with spiking neural P systems.

Device	SNPS Model	Advantages and Disadvantages
1. Traction power supply system [88]	WFRSNPS	Four types of neurons; Expressing status information of protective relays and circuit breakers efficiently; Weighted matrix-based reasoning algorithm (WMBRA);
2. Metro traction power system [82]	MFRSNPS	Three types of neurons; Represent operation information of PRs and CBs efficiently; Reasoning algorithm based on MFRSNPS; Accurately identify faults with certain/uncertain and complete/incomplete operate information.
3. Electric locomotive systems [87]	rFRSNPS	(1) Efficiently expresses Relationships among breakdown signals and faulty sections in subsystems of electric locomotive systems; (2) No experimental data for large scale systems.

Table 7. Comparisons of reasoning algorithms for transformers.

Device	SNPS Model	Features of Reasoning Algorithms
1. Transformers [84]	FRSNPS	Proposition neurons work as input neuron; Based on neuron's firing mechanism and uses matrix operation; Reason out the fuzzy truth values of other unknown fuzzy propositions from known fuzzy propositions (input neurons) Based on FRSNPS; The components of truth value vector are real numbers.
2. Transformers [128]	FRSNPS	Same as [84]
3. Transformers [91]	LSNPS	No reasoning algorithm; Uses Belief AdaBoost Algorithm;
4. Power transmission network [89]	FDSNPS	Proposition neurons work as input neuron; as input neuron; Based on neuron's firing mechanism and uses matrix operation; Reason out the fuzzy truth values of other unknown fuzzy propositions from known fuzzy propositions (input neurons) Based on FDSNPS; Components of truth value vectors are trapezoidal fuzzy numbers;

Table 8. Comparisons of reasoning algorithms for transmission lines.

1. Power transmission network [125]	srSNPS	Self-updating matrix reasoning algorithm; Reason out the fuzzy truth values of other unknown fuzzy propositions from known fuzzy propositions (input neurons) Based on srSNPS Components of truth value vectors are real numbers.
2. Power transmission network [129]	IVFSNPS	Based on IVFSNPS; Synaptic connection between proposition neuron and \odot and \triangle rule neurons represented using matrix and used in algorithm. Components of truth value vector are integer valued fuzzy numbers.
3. Traction power supply system [88]	WFRSNPS	Weighted matrix-based reasoning algorithm (WMBRA) Along with matrix for synaptic connections, there exists synaptic weight matrix. Components of truth value vectors are real numbers.
4. Metro traction power system [82]	MFRSNPS	Based on MFRSNPS; The matrix representing synaptic connections between proposition and General, And, Or rule neuron used. Components of the vectors representing potential values of neurons are real numbers.

Table 9. Comparisons of reasoning algorithms for other types of power systems.

1. Fault section estimation of power systems [126]	OSNPS	Based on OSNPS; Uses guider algorithm and OSNPS fault section estimation algorithm.
2. Fault location distribution network [86]	IASNPS	Based on IASNPS; The values of components of the vectors representing fuzzy truth values of the neurons are $-1, 1, 0$.
3. Fault location distribution network [67]	ESTSNPS	High accuracy, less computation, simple and intuitive model and reasoning. Applied reasonably in the bidirectional power flow characteristics of a distribution network with DGs; Forward reasoning process; Backward verification process Results contained in the neuron are different during the forward reasoning process and the backward verification process; Based on ESTSNPS
4. Fault line detection [85]	rFRSNPS	Based on rFRSNPS; Matrices represent synaptic connections between the neurons; The components of the vector representing truth value of neurons are real numbers.
5. Electric locomotive systems [87]	rFRSNPS	Based on rFRSNPS; Matrices represent synaptic connections between the neurons; The components of the vector representing truth value of neurons are real numbers.

3. Software

The fault diagnosis models based on different variants of SNPS discussed above are performed manually. So it is time consuming as well as very difficult to be used for large scale networks. Next we discuss a method which can perform the task of fault diagnosis automatically.

The FRSNPS model and its variants introduced in [82,84,87–89,125,128,131,132] can efficiently describe the relationship between the faults and protective devices and fault diagnosis model because of the use of mathematical expressions. These models also have good fault tolerant capacity. These intrinsic properties make the FRSNPS models a preferable choice for applications of fault diagnosis. However, one of the major drawback of these models is that the implementation of these models is manual and it prohibits the use of these models to diagnose faults on large power systems networks. In [127], in order to automatically perform this task, an automatic implementation method was introduced. This method is called as *Membrane computing fault diagnosis method* (MCFD) and this method

consists of automatic input and output. Furthermore, the diagnosis process is divided into four sections, i.e., (i) network topology analysis; (ii) suspicious fault component analysis; (iii) construction of FRSNPS models based on (ii); and (iv) fuzzy inference.

Now we give a summary of the MCFD method step by step:

1. **Automatized Input:** Network topology information and protection configuration information is given as input and stored in the access database. These information is also used to construct topology and protection configuration table. Furthermore, topology data of a power system and protection configuration data are given as input to fault diagnosis model based on FRSNPS.

2. **Topology Data of Power Systems:** The topology data is stored into a table which contains the information of the main components and switches. It also contains the connection relationship between the components and switches and the protection number associated with the components.

3. **Protection configuration data:** Whenever there is a fault in power systems, the protective relays and circuit breakers are operated in order to isolate the fault. The information regarding the tripped CBs and PRs is obtained from the SCADA systems and a correlation database is constructed from these information.

4. **Automatized Network Topology Analysis:** Using this method the suspicious fault components are identified. In this method, at first all subsets are searched and then passive networks (outage areas) are identified from these subsets.

5. **Automatized Suspicious Fault Component Analysis:** A logic diagram is constructed where the suspected fault component is the starting point. It also portrays the topological association of the suspected faults components and the protection associated with it in power grid. The logic diagram searches and builds towards each connection from the starting point so that the component remains protected.

6. **Automatized Modeling Suspicious Fault Components with FRSNP System:** According to the fault production rules of the suspected fault components, a FRSNPS model is constructed.

7. **Automatized Fuzzy Inference:** Then reasoning algorithm is performed. It is performed after the confidence level of the proposition, i.e., proposition neuron and CF (certainty factor) of the rule neuron are obtained. Furthermore, after executing the fuzzy reasoning algorithm, output of the proposition neurons is obtained which also represents the fuzzy values of the propositions.

8. **Automatized Output:** The fault component information, protective relay information and circuit breakers operation evaluation are obtained as output of MCFD. Whenever any fault occurs in the system the suspected fault components can be identified from the fuzzy value of the output proposition neuron which is obtained according to the fault threshold associated with it. More specifically, if there is any fault in the component, then the fault component information and fault credibility value are obtained as output. Next from the reasoning process, it is estimated whether any misoperation occurred in the circuit breakers and protective relays. If no misoperation occurred inside the CBs and PRs, then the output appears normal.

4. Conclusions and Future Research Lines

In this paper we provided a comprehensive survey of the membrane computing models, i.e., spiking neural P systems used for fault diagnosis of power systems. We also discussed corresponding reasoning algorithms based on these models. However the investigation regarding the construction of fault diagnosis models based on membrane computing models is still in preliminary stage. Hence there exists many topics which need to be investigated further. We discuss some of the topics which can be considered as future topic of research below.

- (1) The FRSNPS models and their variants have been very useful for fault diagnosis in power systems. Constructing new variants of FRSNPS models by incorporating the properties of neural networks or deep neural networks such that these models have

the capability of handling fuzzy, vague or incomplete information could be a future research direction.

(2) The reasoning algorithms discussed in this paper are efficient. However, there has been no known study discussing the complexity of these algorithms. It is not known whether the time and space complexity of these algorithms are optimal or not. Investigating optimality of these algorithms and constructing reasoning algorithms with better bounds could also be a future research direction.

(3) In this paper we discussed the use of a supervised learning algorithm in the framework of SNPS for identification of faults in transformers. However, it remains to be investigated whether any unsupervised learning algorithm in the framework of SNPS can be constructed such that it can be used for fault diagnosis in power systems.

(4) The investigations on the use of FRSNPS models for power systems fault diagnosis in the literature mentioned in this paper indicate that FRSNPS models have the potential to diagnose faults in large-scale power systems with a linear or polynomial time. However, the proof or experimental verification is a future work.

Author Contributions: Conceptualization, Y.L. and G.Z.; investigation, P.P. and Y.L.; writing—original draft preparation, P.P.; writing—review and editing, Y.L., P.P., Y.C., S.F., X.M. and G.Z.; supervision, G.Z.; funding acquisition, Y.L., Y.C., S.F. and X.M. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Science and Technology Project of State Grid Sichuan Electric Power Company (No. 521997180016).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

SNPS	Spiking Neural P System
SCADA	Supervisor Control And Data Control
FRSNPS	Fuzzy Reasoning Spiking Neural P Systems
DGA	Dissolved Gas Analysis
LSNPS	Learning Spiking Neural P System
NN	Neural Network
SVM	Support Vector Machine
DBN	Deep Belief Network
FDSNPS	FRSNPS with Trapezoidal Fuzzy Numbers
WFRSNPS	Weighted Fuzzy Reasoning Spiking Neural P Systems
TPSSs	Traction Power Supply System
PRs	Protective Relays
CBs	Circuit Breakers
MTPSSs	Metro Traction Power Supply Systems
OSNPS	Optimization Spiking Neural P Systems
FSE	Fault Section Estimation
ESNPS	Extended Spiking Neural P Systems
IASNPS	Improved Spiking Neural P Systems with Anti-Spikes
rFRSNPS	Fuzzy Reasoning Spiking Neural P Systems with real numbers
MCFD	Membrane Computing Fault Diagnosis method

References

- Chien, C.F.; Chen, S.L.; Lin, Y.S. Using Bayesian network for fault location on distribution feeder. *IEEE Trans. Power Deliv.* **2002**, *17*, 785–793.
- Zhu, Y.L.; Huo, L.M.; Liu, J.L. Bayesian networks based approach for power systems fault diagnosis. *IEEE Trans. Power Deliv.* **2006**, *21*, 634–639. [[CrossRef](#)]
- Cardoso, G.; Rolim, J.G.; Zurn, H.H. Identifying the primary fault section after contingencies in bulk power systems. *IEEE Trans. Power Deliv.* **2008**, *23*, 1335–1342. [[CrossRef](#)]
- Thukaram, D.; Khincha, H.P.; Vijaynarasimha, H.P. Artificial neural network and support vector machine approach for locating faults in radial distribution systems. *IEEE Trans. Power Deliv.* **2005**, *20*, 710–721. [[CrossRef](#)]

5. Wen, F.S.; Han, Z.X. Fault section estimation in power systems using a genetic algorithm. *Electr. Power Syst. Res.* **1995**, *34*, 165–172. [[CrossRef](#)]
6. Jiang, Z.; Li, Z.; Wu, N.; Zhou, M. A Petri net approach to fault diagnosis and restoration for power transmission systems to avoid the output interruption of substations. *IEEE Syst. J.* **2018**, *12*, 2566–2576. [[CrossRef](#)]
7. Luo, X.; Kezunovic, M. Implementing fuzzy reasoning Petri-nets for fault section estimation. *IEEE Trans. Power Syst.* **2008**, *23*, 676–685. [[CrossRef](#)]
8. Sun, J.; Qin, S.Y.; Song, Y.H. Fault diagnosis of electric power systems based on fuzzy Petri nets. *IEEE Trans. Power Syst.* **2004**, *19*, 2053–2059.
9. Yang, J.W.; He, Z.Y.; Zang, T.L. Power system fault-diagnosis method based on directional weighted fuzzy Petri nets. *Proc. CSEE* **2010**, *30*, 42–49.
10. Lee, H.J.; Ahn, B.S.; Park, Y.M. A fault diagnosis expert system for distribution substations. *IEEE Trans. Power Deliv.* **2008**, *15*, 92–97. [[CrossRef](#)]
11. Ma, D.Y.; Liang, Y.C.; Zhao, X.S.; Guan, R.C.; Shi, X.H. Multi-BP expert system for fault diagnosis of power system. *Eng. Appl. Artif. Intell.* **2013**, *26*, 937–944. [[CrossRef](#)]
12. Chang, C.S.; Chen, J.M.; Srinivasan, D.; Wen, F.S.; Liew, A.C. Fuzzy logic approach in power system fault section identification. *IEE Proc. Gener. Transm. Distrib.* **1997**, *144*, 406–414. [[CrossRef](#)]
13. Davidson, E.M.; McArthur, S.D.J.; McDonald, J.R.; Watt, I. Applying multi-agent system technology in practice: Automated management and analysis of SCADA and digital fault recorder data. *IEEE Trans. Power Syst.* **2006**, *21*, 559–567. [[CrossRef](#)]
14. Hossack, J.A.; Menal, J.; McArthur, S.D.J.; McDonald, J.R. A multiagent architecture for protection engineering diagnostic assistance. *IEEE Trans. Power Syst.* **2003**, *18*, 639–647.
15. He, Z.Y.; Chiang, H.D.; Li, C.W.; Zeng, Q.F. Fault-section estimation in power systems based on improved optimization model and binary pswarm optimization. In Proceedings of the 2009 IEEE Power & Energy Society General Meeting, Calgary, AB, Canada, 26–30 July 2009; pp. 1–8. [[CrossRef](#)]
16. Lin, X.N.; Ke, S.H.; Li, Z.T.; Weng, H.L.; Han, X.H. A fault diagnosis method of power systems based on improved objective function and genetic algorithm-tabu search. *IEEE Trans. Power Deliv.* **2010**, *25*, 1268–1274.
17. Lin, S.; He, Z.Y.; Qian, Q.Q. Review and development on fault diagnosis in power grid. *Power Syst. Protect. Control* **2010**, *38*, 140–150.
18. Tang, L.; Sun, H.B.; Zhang, B.M.; Gao, F. Online fault diagnosis for power system based on information theory. *Proc. CSEE* **2003**, *23*, 5–11. [[CrossRef](#)]
19. Chen, W.H. Fault section estimation using fuzzy matrix-based reasoning methods. *IEEE Trans. Power Deliv.* **2011**, *26*, 205–213. [[CrossRef](#)]
20. Chen, W.H. Online fault diagnosis for power transmission networks using fuzzy digraph models. *IEEE Trans. Power Deliv.* **2012**, *27*, 688–698. [[CrossRef](#)]
21. Păun, G. Computing with Membranes. *J. Comput. Syst. Sci.* **2000**, *61*, 108–143.
22. Zhang, G.; Pérez-Jiménez, M.J.; Riscos-Núñez, A.; Verlan, S.; Konur, S.; Hinze, T.; Gheorghe, M. *Membrane Computing Models: Implementations*; Springer: Singapore, 2021. [[CrossRef](#)]
23. Gazdag, Z.; Kolonits, G. A new method to simulate restricted variants of polarizationless P systems with active membranes. *J. Membr. Comput.* **2019**, *1*, 251–261. [[CrossRef](#)]
24. Orellana-Martín, D.; Valencia-Cabrera, L.; Riscos-Núñez, A.; Pérez-Jiménez, M.J. Minimal cooperation as a way to achieve the efficiency in cell-like membrane systems. *J. Membr. Comput.* **2019**, *1*, 85–92. [[CrossRef](#)]
25. Orellana-Martín, D.; Valencia-Cabrera, L.; Riscos-Núñez, A.; Pérez-Jiménez, M.J. P systems with proteins: A new frontier when membrane division disappears. *J. Membr. Comput.* **2019**, *1*, 29–39. [[CrossRef](#)]
26. Buno, K.; Adorna, H. Distributed computation of a k P systems with active membranes for SAT using clause completion. *J. Membr. Comput.* **2020**, *2*, 108–120. [[CrossRef](#)]
27. Henderson, A.; Nicolescu, R.; Dinneen, M.J. Solving a PSPACE-complete problem with cP systems. *J. Membr. Comput.* **2020**, *2*, 311–322. [[CrossRef](#)]
28. Orellana-Martín, D.; Riscos-Núñez, A. Seeking computational efficiency boundaries: The Păun’s conjecture. *J. Membr. Comput.* **2020**, *2*, 323–331. [[CrossRef](#)]
29. Leporati, A.; Manzoni, L.; Mauri, G.; Porreca, A.E.; Zandron, C. A Turing machine simulation by P systems without charges. *J. Membr. Comput.* **2020**, *2*, 71–79. [[CrossRef](#)]
30. Sosík, P. P systems attacking hard problems beyond NP: A survey. *J. Membr. Comput.* **2019**, *1*, 198–208. [[CrossRef](#)]
31. Alhazov, A.; Freund, R.; Ivanov, S. P systems with limited number of objects. *J. Membr. Comput.* **2020**, *3*, 1–9. [[CrossRef](#)]
32. Jiang, Y.; Su, Y.; Luo, F. An improved universal spiking neural P system with generalized use of rules. *J. Membr. Comput.* **2019**, *1*, 270–278. [[CrossRef](#)]
33. Păun, G.; Rozenberg, G.; Salomaa, A. *The Oxford Handbook of Membrane Computing*; Oxford University Press: London, UK, 2009.
34. Leporati, A.; Manzoni, L.; Mauri, G.; Porreca, A.E.; Zandron, C. Characterizing PSPACE with shallow non-confluent P systems. *J. Membr. Comput.* **2019**, *1*, 75–84. [[CrossRef](#)]
35. Zandron, C. Bounding the space in P systems with active membranes. *J. Membr. Comput.* **2020**, *2*, 137–145. [[CrossRef](#)]

36. Leporati, A.; Manzoni, L.; Mauri, G.; Porreca, A.E.; Zandron, C. Shallow laconic P systems can count. *J. Membr. Comput.* **2020**, *2*, 49–58. [[CrossRef](#)]
37. Leporati, A.; Manzoni, L.; Mauri, G.; Porreca, A.E.; Zandron, C. Simulating counting oracles with cooperation. *J. Membr. Comput.* **2020**, *2*, 303–310. [[CrossRef](#)]
38. Juayong, R.A.B.; Adorna, H.N. A survey of results on evolution—Communication P systems with energy. *J. Membr. Comput.* **2020**, *2*, 59–69.
39. Zhang, G.; Pérez-Jiménez, M.J.; Gheorghe, M. *Real-Life Applications with Membrane Computing, Emergence, Complexity and Computation Book Series (ECC)*; Springer: New York, NY, USA, 2017; Volume 25. [[CrossRef](#)]
40. Andreu-Guzmán, J.A.; Valencia-Cabrera, L. A novel solution for GCP based on an OLMS membrane algorithm with dynamic operators. *J. Membr. Comput.* **2020**, *2*, 1–13. [[CrossRef](#)]
41. Zhang, G.; Gheorghe, M.; Pan, L.; Pérez-Jiménez, M.J. Evolutionary membrane computing: A comprehensive survey and new results. *Inf. Sci.* **2014**, *279*, 528–551. [[CrossRef](#)]
42. Zhang, G.; Cheng, J.; Gheorghe, M.; Meng, Q. A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems. *Appl. Soft Comput.* **2013**, *13*, 1528–1542.
43. Zhang, G.X.; Gheorghe, M.; Wu, C.Z. A quantum-inspired evolutionary algorithm based on P systems for knapsack problem. *Fundam. Inform.* **2008**, *87*, 93–116.
44. Ou, Z.; Zhang, G.; Wang, T.; Huang, X. Automatic design of cell-like P systems through tuning membrane structures, initial objects and evolution rules. *Int. J. Unconv. Comput.* **2013**, *9*, 425–443. [[CrossRef](#)]
45. Yuan, W.; Zhang, G.; Pérez-Jiménez, M.J.; Wang, T.; Huang, Z. P systems based computing polynomials: Design and formal verification. *Nat. Comput.* **2016**, *15*, 591–596. [[CrossRef](#)] [[PubMed](#)]
46. Zhang, G.; Rong, H.; Ou, Z.; Pérez-Jiménez, M.J.; Gheorghe, M. Automatic design of deterministic and non-halting membrane systems by tuning syntactical ingredient. *IEEE Trans. NanoBiosci.* **2014**, *13*, 363–371. [[CrossRef](#)] [[PubMed](#)]
47. Zhu, M.; Zhang, G.; Yang, Q.; Rong, H.; Yuan, W.; Pérez-Jiménez, M.J. P systems based computing polynomials with integer coefficients: Design and formal verification. *IEEE Trans. NanoBiosci.* **2018**, *17*, 272–280. [[CrossRef](#)]
48. Buiu, C.; Florea, A.G. Membrane computing models and robot controller design, current results and challenges. *J. Membr. Comput.* **2019**, *1*, 262–269. [[CrossRef](#)]
49. Díaz-Pernil, D.; Gutiérrez-Naranjo, M.A.; Peng, H. Membrane computing and image processing: A short survey. *J. Membr. Comput.* **2019**, *1*, 58–73. [[CrossRef](#)]
50. Mayne, R.; Phillips, N.; Adamatzky, A. Towards experimental P-systems using multivesicular liposomes. *J. Membr. Comput.* **2019**, *1*, 20–28. [[CrossRef](#)]
51. Sánchez-Karhunen, E.; Valencia-Cabrera, L. Modelling complex market interactions using PDP systems. *J. Membr. Comput.* **2019**, *1*, 40–51. [[CrossRef](#)]
52. Zhu, M.; Yang, Q.; Dong, J.; Zhang, G.; Gou, X.; Rong, H.; Paul, P.; Neri, F. An adaptive optimization spiking neural P system for binary problems. *Int. J. Neural Syst.* **2021**, *31*, 2050054. [[CrossRef](#)]
53. Perez-Hurtado, I.; Martinez-del-Amor, M.A.; Zhang, G.; Neri, F.; Pérez-Jiménez, M.J. A membrane parallel rapidly-exploring random tree algorithm for robotic motion planning. *Integr. Comput. Aided Eng.* **2020**, *27*, 121–138. [[CrossRef](#)]
54. Wang, X.; Zhang, G.; Gou, X.; Paul, P.; Neri, F.; Rong, H.; Yang, Q.; Zhang, H. Multi-behaviors coordination controller design with enzymatic numerical P systems for robots. *Integr. Comput. Aided Eng.* **2021**, *28*, 119–140. [[CrossRef](#)]
55. Wang, X.; Zhang, G.; Neri, F.; Jiang, T.; Zhao, J.; Gheorghe, M.; Ipate, F.; Lefticaru, R. Design and implementation of membrane controllers for trajectory tracking of nonholonomic wheeled mobile robots. *Integr. Comput. Aided Eng.* **2016**, *23*, 15–30. [[CrossRef](#)]
56. Ochirbat, O.; Ishdorj, T.-O.; Cichon, G. An error-tolerant serial binary full-adder via a spiking neural P system using HP/LP basic neurons. *J. Membr. Comput.* **2020**, *2*, 42–48. [[CrossRef](#)]
57. Ceterchi, R.; Subramanian, K.G. Generating pictures in string representation with P systems: The case of space-filling curves. *J. Membr. Comput.* **2020**, *2*, 369–379. [[CrossRef](#)]
58. Cienialová, L.; Csuhaaj-Varjú, E.; Cieniala, L.; Sosík, P. P colonies. *J. Membr. Comput.* **2019**, *1*, 178–197. [[CrossRef](#)]
59. Cienialová, L.; Cieniala, L.; Csuhaaj-Varjú, E. P colonies and reaction systems. *J. Membr. Comput.* **2020**, *2*, 269–280. [[CrossRef](#)]
60. García-Quismondo, M.; Hintz, W.D.; Schuler, M.S.; Relyea, R.A. Modeling diel vertical migration with membrane computing. *J. Membr. Comput.* **2021**, *3*, 35–50. [[CrossRef](#)]
61. Nash, A.; Kalvala, S. A P system model of swarming and aggregation in a Myxobacterial colony. *J. Membr. Comput.* **2019**, *1*, 103–111.
62. Ionescu, M.; Păun, G.; Yokomori, T. Spiking neural P systems. *Fundam. Inform.* **2006**, *71*, 279–308. [[CrossRef](#)]
63. Maass, W. Networks of spiking neurons: The third generation of neural network models. *Neural Netw.* **1997**, *10*, 1659–1671. [[CrossRef](#)]
64. De la Cruz, R.T.A.; Cabarle, F.G.C.; Adorna, H.N. Generating context-free languages using spiking neural P systems with structural plasticity. *J. Membr. Comput.* **2019**, *1*, 161–177. [[CrossRef](#)]
65. Song, T.; Pan, L. Spiking neural P systems with request rules. *Neurocomputing* **2016**, *193*, 193–200. [[CrossRef](#)]
66. Liu, X.; Li, Z.; Suo, J.; Liu, J.; Min, X. A uniform solution to integer factorization using time-free spiking neural P system. *Neural Comput. Appl.* **2015**, *26*, 1241–1247. [[CrossRef](#)]

67. Sun, Z.; Wang, Q.; Wei, Z. Fault location of distribution network with distributed generations using electrical synaptic transmission-based spiking neural P systems. *Int. J. Parallel Emergent Distrib. Syst.* **2021**, *36*, 11–27. [[CrossRef](#)]
68. Chen, K.; Wang, J.; Sun, Z.; Luo, J.; Liu, T. Programmable Logic Controller Stage Programming Using Spiking Neural P Systems. *J. Comput. Theor. Nanosci.* **2015**, *12*, 1292–1299. [[CrossRef](#)]
69. Huang, K.; Wang, T.; He, Y.; Zhang, G.; Pérez-Jiménez, M.J. Temporal fuzzy reasoning spiking neural P systems with real numbers for power system fault diagnosis. *J. Comput. Theor. Nanosci.* **2016**, *13*, 3804–3814. [[CrossRef](#)]
70. Jimenez, Z.B.; Cabarle, F.G.C.; de la Cruz, R.T.A.; Buño, K.C.; Adorna, H.N.; Hernandez, N.H.S.; Zeng, X. Matrix representation and simulation algorithm of spiking neural P systems with structural plasticity. *J. Membr. Comput.* **2019**, *1*, 145–160. [[CrossRef](#)]
71. Bao, T.; Zhou, N.; Lv, Z.; Peng, H.; Wang, J. Sequential dynamic threshold neural P systems. *J. Membr. Comput.* **2020**, *2*, 255–268.
72. Cavaliere, M.; Egecioglu, O.; Ibarra, O.H.; Woodworth, S.; Ionescu, M.; Păun, G. *Asynchronous Spiking Neural P Systems: Decidability and Undecidability*; Garzon, M.H., Yan, H., Eds.; Springer: Berlin, Germany, 2008; pp. 246–255. [[CrossRef](#)]
73. Pan, L.; Wang, J.; Hoogeboom, H. Spiking neural P systems with astrocytes. *Neural Comput.* **2012**, *24*, 805–825. [[CrossRef](#)]
74. Su, Y.; Wu, T.; Xu, F.; Păun, A. Spiking Neural P Systems with Rules on Synapses Working in Sum Spikes Consumption Strategy. *Fundam. Inform.* **2017**, *156*, 187–208. [[CrossRef](#)]
75. Pan, L.; Păun, G.; Zhang, G.; Neri, F. Spiking neural P systems with communication on request. *Int. J. Neural Syst.* **2017**, *27*, 1750042. [[CrossRef](#)]
76. Cabarle, F.G.C.; Adorna, H.N.; Jiang, M.; Zeng, X. Spiking Neural P Systems With Scheduled Synapses. *IEEE Trans. Nanobiosci.* **2017**, *16*, 792–801. [[CrossRef](#)]
77. Cabarle, F.G.C.; Adorna, H.N.; Pérez-Jiménez, M.J.; Song, T. Spiking neural P systems with structural plasticity. *Neural Comput. Appl.* **2015**, *26*, 1905–1917. [[CrossRef](#)]
78. De la Cruz, R.T.A.; Cabarle, F.G.; Macababayao, I.C.H.; Adorna, H.N.; Zeng, X. Homogeneous spiking neural P systems with structural plasticity. *J. Membr. Comput.* **2021**, *3*, 10–21. [[CrossRef](#)]
79. Pan, L.; Zeng, X.; Zhang, X.; Jiang, Y. Spiking Neural P Systems with Weighted Synapses. *Neural Process. Lett.* **2012**, *35*, 13–27. [[CrossRef](#)]
80. Song, T.; Wang, X. Homogenous Spiking Neural P Systems with Inhibitory Synapses. *Neural Process. Lett.* **2015**, *42*, 199–214. [[CrossRef](#)]
81. Pan, L.; Păun, G. Spiking neural P systems with anti-spikes. *Int. J. Comput. Commun. Control.* **2009**, *4*, 273–282.
82. He, Y.; Wang, T.; Huang, K.; Zhang, G.; Pérez-Jiménez, M.J. Fault diagnosis of metro traction power systems using a modified fuzzy reasoning spiking neural P system. *Rom. J. Inf. Sci. Technol.* **2015**, *18*, 256–272. [[CrossRef](#)]
83. Huang, Y.; Wang, T.; Wang, J.; Peng, H. Reliability evaluation of distribution network based on fuzzy spiking neural P system with self-synapse. *J. Membr. Comput.* **2021**, *3*, 51–62. [[CrossRef](#)]
84. Peng, H.; Wang, J.; Pérez-Jiménez, M.J.; Wang, H.; Shao, J.; Wang, T. Fuzzy reasoning spiking neural P system for fault diagnosis. *Inf. Sci.* **2013**, *235*, 106–116. [[CrossRef](#)]
85. Rong, H.; Ge, M.; Zhang, G.; Zhu, M. An Approach for Detecting Fault Lines in a Small Current Grounding System using Fuzzy Reasoning Spiking Neural P Systems. *Int. J. Comput. Commun. Control.* **2018**, *13*, 521–536.
86. Tao, C.; Wang, J.; Wang, T.; Yang, Y. An Improved Spiking Neural P Systems with Anti-Spikes for Fault Location of Distribution Networks with Distributed Generation. In *Bio-Inspired Computing: Theories and Applications. BIC-TA 2017. Communications in Computer and Information Science*; He, C., Mo, H., Pan, L., Zhao, Y., Eds.; Springer: Singapore, 2017; Volume 791.
87. Wang, T.; Zhang, G.; Pérez-Jiménez, M.J. Fault Diagnosis Models for Electric Locomotive Systems Based on Fuzzy Reasoning Spiking Neural P Systems. In *Membrane Computing*; Gheorghe, M., Rozenberg, G., Salomaa, A., Sosík, P., Zandron, C., Eds.; CMC 2014, LNCS 8961; Springer: New York, NY, USA, 2014. [[CrossRef](#)]
88. Wang, T.; Zhang, G.; Pérez-Jiménez, J.M.; Cheng, J. Weighted fuzzy reasoning spiking neural P systems: Application to fault diagnosis in traction power supply systems of high-speed railways. *J. Comput. Theor. Nanosci.* **2015**, *12*, 1103–1114. [[CrossRef](#)]
89. Wang, T.; Zhang, G.; Zhao, J.; He, Z.; Wang, J.; Pérez-Jiménez, M.J. Fault diagnosis of electric power systems based on fuzzy reasoning spiking neural P systems. *IEEE Trans. Power Syst.* **2015**, *30*, 1182–1194. [[CrossRef](#)]
90. Wang, T.; Zhang, G.; Pérez-Jiménez, M.J. Fuzzy membrane computing: Theory and Applications. *Int. J. Comput. Comm. Contr.* **2015**, *10*, 904–935.
91. Zhang, X.; Zhang, G.; Paul, P.; Zhang, J.; Wu, T.; Fan, S.; Xiong, X. Dissolved Gas Analysis for Transformer Fault Based on Learning Spiking Neural P System with Belief AdaBoost. *Int. J. Unconv. Comput.* **2021**, *16*, 239–258. [[CrossRef](#)]
92. Ma, T.; Hao, S.; Wang, X.; Rodríguez-Patón, A.; Wang, S.; Song, T. Double Layers Self-Organized Spiking Neural P Systems with Anti-spikes for Fingerprint Recognition. *IEEE Access* **2017**, *7*, 177562–177570. [[CrossRef](#)]
93. Song, T.; Pan, L.; Wu, T.; Zheng, P.; Wong, M.L.D.; Rodríguez-Patón, A. Spiking Neural P Systems With Learning Functions. *IEEE Trans. Nanobiosci.* **2019**, *18*, 176–190. [[CrossRef](#)]
94. Song, T.; Pang, S.; Hao, S.; Rodríguez-Patón, A.; Zheng, P. A Parallel Image Skeletonizing Method Using Spiking Neural P Systems with Weights. *Neural Process. Lett.* **2019**, *50*, 1485–1502. [[CrossRef](#)]
95. Chen, Z.; Zhang, P.; Wang, X.; Shi, X.; Wu, T.; Zheng, P. A computational approach for nuclear export signals identification using spiking neural P systems. *Neural Comput. Appl.* **2018**, *29*, 695–705. [[CrossRef](#)]
96. Díaz, C.; Sanchez, G.; Duchon, G.; Nakano, M.; Perez, H. An efficient hardware implementation of a novel unary Spiking Neural Network multiplier with variable dendritic delays. *Neurocomputing* **2016**, *189*, 130–134. [[CrossRef](#)]

97. Duchen, G.; Diaz, C.; Sanchez, G.; Perez, H. First steps toward memory processor unit architecture based on SN P systems. *Electron. Lett.* **2017**, *53*, 384–385. [[CrossRef](#)]
98. Ganbaatar, G.; Nyamdorj, D.; Cichon, G.; Ishdorj, T. Implementation of RSA cryptographic algorithm using SN P systems based on HP/LP neurons. *J. Membr. Comput.* **2021**, *3*, 22–34.
99. Gutiérrez-Naranjo, M.A.; Loporati, A. Performing arithmetic operations with spiking neural P systems. In Proceedings of the 7th Brainstorming Week Membrane Computing, Sevilla, Spain, 27 February–2 April 2009; pp. 181–198. [[CrossRef](#)] [[PubMed](#)]
100. Liu, X.; Li, Z.; Liu, J.; Liu, L.; Zeng, X. Implementation of arithmetic operations with time-free spiking neural P systems. *IEEE Trans. Nanobiosci.* **2015**, *14*, 617–624. [[CrossRef](#)] [[PubMed](#)]
101. Zeng, X.; Song, T.; Zhang, X.; Pan, L. Performing four basic arithmetic operations with spiking neural P systems. *IEEE Trans. Nanobiosci.* **2012**, *11*, 366–374. [[CrossRef](#)] [[PubMed](#)]
102. Zhang, G.; Rong, H.; Paul, P.; He, Y.; Neri, F.; Pérez-Jiménez, M.J. A Complete Arithmetic Calculator Constructed from Spiking Neural P Systems and its Application to Information Fusion. *Int. J. Neural Syst.* **2021**, *31*, 2050055. [[CrossRef](#)]
103. Zhang, G.; Shang, Z.; Verlan, S.; Martínez-Del-Amor, M.A.; Yuan, C.; Valencia-Cabrera, L.; Pérez-Jiménez, M.J. An overview of hardware implementation of membrane computing models. *ACM Comput. Surv.* **2020**, *53*, 90. [[CrossRef](#)]
104. Ishdorj, T.O.; Ochirbat, O.; Naimannaran, C. A μ -fluidic Biochip Design for Spiking Neural P Systems. *Int. J. Unconv. Comput.* **2020**, *15*, 59–82.
105. Li, J.; Huang, Y.; Xu, J. Decoder Design Based on Spiking Neural P Systems. *IEEE Trans. Nanobiosci.* **2016**, *15*, 639–644. [[CrossRef](#)]
106. Song, T.; Zheng, P.; Wong, M.L.D.; Wang, X. Design of logic gates using spiking neural P systems with homogeneous neurons and astrocytes-like control. *Inf. Sci.* **2016**, *372*, 380–391. [[CrossRef](#)]
107. Adorna, H.N. Computing with SN P systems with I/O mode. *J. Membr. Comput.* **2020**, *2*, 230–245. [[CrossRef](#)]
108. Valencia-Cabrera, L.; Orellana-Martín, D.; Martínez-del-Amor, M.A.; Pérez-Jiménez, M.J. An interactive timeline of simulators in membrane computing. *J. Membr. Comput.* **2019**, *1*, 209–222. [[CrossRef](#)]
109. Pérez-Hurtado, I.; Orellana-Martín, D.; Zhang, G.; Pérez-Jiménez, M.J. P-Lingua in two steps: Flexibility and efficiency. *J. Membr. Comput.* **2019**, *1*, 93–102.
110. García-Quismondo, M.; Gutiérrez-Escudero, R.; Pérez-Hurtado, I.; Pérez-Jiménez, M.J.; Riscos-Núñez, A. An Overview of P-Lingua 2.0. In *Membrane Computing*; Păun, G., Pérez-Jiménez, M.J., Riscos-Núñez, A., Rozenberg, G., Salomaa, A., Eds.; WMC 2009, Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2010; Volume 5957.
111. Macari, V.; Magariu, G.; Verlan, T. Simulator of P-Systems with String Replication Developed in Framework of P-Lingua 2.1. *Comput. Sci. J. Moldova* **2010**, *18*, 246–268. [[CrossRef](#)]
112. Martínez-del-Amor, M.A.; Pérez-Hurtado, I.; Pérez-Jiménez, M.J.; Riscos-Núñez, A. A P-Lingua based simulator for tissue P systems. *J. Logic Algebraic Program.* **2010**, *79*, 374–382.
113. Valencia-Cabrera, L.; Wu, T.; Zhang, Z.; Pan, L.; Pérez-Jiménez, M.J. A Simulation Software Tool for Cell-like Spiking Neural P Systems. *Rom. J. Inf. Sci. Technol.* **2016**, *20*, 71–84.
114. Macías-Ramos, L.F.; Pérez-Hurtado, I.; García-Quismondo, M.; Valencia-Cabrera, L.; Pérez-Jiménez, M.J.; Riscos-Núñez, A. A P-Lingua Based Simulator for Spiking Neural P Systems. In *Membrane Computing*; Gheorghe, M., Păun, G., Rozenberg, G., Salomaa, A., Verlan, S., Eds.; CMC 2011, Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7184. [[CrossRef](#)]
115. Macías-Ramos, L.F.; Pérez-Jiménez, M.J.; Song, T.; Pan, L. Extending Simulation of Asynchronous Spiking Neural P Systems in P-Lingua. *Fundam. Inform.* **2015**, *136*, 253–267. [[CrossRef](#)]
116. Valencia-Cabrera, L.; Song, B. Tissue P systems with promoter simulation with MeCoSim and P-Lingua framework. *J. Membr. Comput.* **2020**, *2*, 95–107.
117. Paul, J.; Carandang, A.; Villaflores, J.M.B.; Cabarle, F.G.C.; Adorna, H.N.; Martínez-del-Amor, M.A. CuSNP: Spiking Neural P Systems Simulators in CUDA. *Rom. J. Inf. Sci. Technol.* **2017**, *20*, 57–70. [[CrossRef](#)]
118. Guo, P.; Quana, C.; Yea, L. UPSimulator: A general P system simulator. *Knowl. Based Syst.* **2019**, *170*, 20–25.
119. Bianco, L.; Manca, V.; Marchetti, L.; Petterlini, M. Psim: A simulator for biomolecular dynamics based on P systems. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 883–887.
120. Arsene, O.; Buiu, C.; Popescu, N. SNUPS: A Simulator for Numerical Membrane Computing. *Int. J. Innov. Comput. Inf. Control* **2011**, *7*, 3509–3522. [[CrossRef](#)]
121. Raghavan, S.; Rai, S.S.; Rohit, M.P.; Chandrasekaran, K. GPUPeP: Parallel Enzymatic Numerical P System simulator with a Python-based interface. *Biosystems* **2020**, *196*, 104186. [[CrossRef](#)]
122. Valencia-Cabrera, L.; Pérez-Hurtado, I.; Martínez-del-Amor, M.A. Simulation challenges in membrane computing. *J. Membr. Comput.* **2020**, *2*, 392–402. [[CrossRef](#)]
123. Cooper, J.; Nicolescu, R. Alternative representations of P systems solutions to the graph colouring problem. *J. Membr. Comput.* **2019**, *1*, 112–126. [[CrossRef](#)]
124. Turlea, A.; Gheorghe, M.; Ipate, F.; Konur, S. Search-based testing in membrane computing. *J. Membr. Comput.* **2019**, *1*, 241–250. [[CrossRef](#)]
125. Liu, W.; Wang, T.; Zang, T.; Huang, Z.; Wang, J.; Huang, T.; Wei, X.; Li, C. A Fault Diagnosis Method for Power Transmission Networks Based on Spiking Neural P Systems with Self-Updating Rules considering Biological Apoptosis Mechanism. *Complexity* **2020**, *2020*, 2462647.

126. Wang, T.; Zeng, S.; Zhang, G.; Pérez-Jiménez, M.J.; Wang, J. Fault Section Estimation of Power Systems with Optimization Spiking Neural P Systems. *Rom. J. Inf. Sci. Technol.* **2015**, *18*, 240–255. [[CrossRef](#)]
127. Rong, H.; Yi, K.; Zhang, G.; Dong, J.; Paul, P.; Huang, Z. Automatic implementation of fuzzy reasoning spiking neural P systems for diagnosing faults in complex power systems. *Complexity* **2019**, *2019*, 2635714. [[CrossRef](#)]
128. Yahya, Y.; Qian, A.; Yahya, A. Power Transformer Fault Diagnosis Using Fuzzy Reasoning Spiking Neural P Systems. *J. Intell. Learn. Syst. Appl.* **2016**, *8*, 77–91. [[CrossRef](#)]
129. Wang, J.; Peng, H.; Yu, W.; Ming, J.; Pérez-Jiménez, M.J.; Tao, C.; Huang, X. Interval-valued fuzzy spiking neural P systems for fault diagnosis of power transmission networks. *Eng. Appl. Artif. Intel.* **2019**, *82*, 102–109. [[CrossRef](#)]
130. Zhang, G.; Rong, H.; Neri, F.; Pérez-Jiménez, M.J. An optimization spiking neural P system for approximately solving combinatorial optimization problems. *Int. J. Neural Syst.* **2014**, *24*, 1440006.
131. Rong, H.; Zhu, M.; Feng, Z.; Zhang, G.; Huang, K. A Novel Approach to Fault Classification of Power Transmission Lines Using Singular Value Decomposition and Fuzzy Reasoning Spiking Neural P Systems. *Rom. J. Inf. Sci. Technol.* **2017**, *20*, 18–31.
132. Tao, C.; Yu, W.; Wang, J.; Peng, H.; Chen, K.; Ming, J. Fault diagnosis of power systems based on triangular fuzzy spiking neural P systems. *Bio-Inspired Comput. Theor. Appl.* **2017**, *618*, 385–398.