IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 August 2022, accepted 6 November 2022, date of publication 2 December 2022, date of current version 15 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3226447

SURVEY

On Memristors for Enabling Energy Efficient and Enhanced Cognitive Network Functions

SAAD SALEH AND BORIS KOLDEHOFE
Bernoulli Institute, University of Groningen, 9700 AK Groningen, The Netherlands
Groningen Cognitive Systems and Materials Center, University of Groningen, 9700 AK Groningen, The Netherlands

Corresponding author: Saad Saleh (s.saleh@rug.nl)

This work was supported by the CogniGron Research Center and the Ubbo Emmius Funds (University of Groningen).

ABSTRACT The high performance requirements of nowadays computer networks are limiting their ability to support important requirements of the future. Two important properties essential in assuring cost-efficient computer networks and supporting new challenging network scenarios are operating energy efficient and supporting cognitive computational models. These requirements are hard to fulfill without challenging the current architecture behind network packet processing elements such as routers and switches. Notably, these are currently dominated by the use of traditional transistor-based components. In this article, we contribute with an in-depth analysis of alternative architectural design decisions to improve the energy footprint and computational capabilities of future network packet processors by shifting from transistor-based components to a novel component named *Memristor*. A memristor is a computational component characterized by non-volatile operations on a physical state, mostly represented in form of (electrical) resistance. Its state can be read or altered by input signals, e.g. electrical pulses, where the future state always depends on the past state. Unlike in traditional von Neumann architectures, the principles behind memristors impose that memory operations and computations are inherently colocated. In combination with the non-volatility, this allows to build memristors at nanoscale size and significantly reduce the energy consumption. At the same time, memristors appear to be highly suitable to model cognitive functionality due to the state dependence transitions in the memristor. In cognitive architectures, our survey contributes to the study of memristor-based Ternary Content Addressable Memory (TCAM) used for storage of cognitive rules inside packet processors. Moreover, we analyze the memristor-based novel cognitive computational architectures built upon self-learning capabilities by harnessing from non-volatility and state-based response of memristors (including reconfigurable architectures, reservoir computation architectures, neural network architectures and neuromorphic computing architectures).

INDEX TERMS Memristors, network architectures, cognitive networks, ternary content addressable memory, neural networks, neuromorphic computing.

I. INTRODUCTION

Nowadays the Internet is the backbone of most computing systems and computer-based applications spanning over server ends, packet processing nodes and users' end devices. The Internet depends on highly efficient packet processing elements optimized to yield high throughput with high performance operations for packet processing, in particular for matching and modifying packet headers, and generating new packets based upon the matches. Modern packet processors perform computational steps at time scales which are in the order of nanoseconds with throughput in the order of terabits e.g., 25.6 Tbps throughput for Intel's programmable Tofino-3 switch [1]. These high performance requirements combined with complex functionalities come at the price of a very high energy footprint. In fact, the energy consumption in transporting data between source and destination in the Internet is 260-340 TWh, which is at least 260, 000 times higher than the average electricity generation by a nuclear power plant [2], [3]. At the time to support requirements on

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Quan.

performance, it also imposes challenges to integrate, change the mechanisms which would allow to utilize the network resources more efficiently. The integration of cognitive functions uptil now is only feasible by connecting external computational elements with packet processors at the cost of high energy consumption and excessive data movement delays. It is highly challenging to integrate these functions at the level/speed of packet processing within the architectures of current network packet processors. Overcoming these challenges requires to fundamentally rethink the design and architecture of network elements.

A particular problem inherent at current design of network elements is the dependence on transistor-based components which have high power consumption and lack non-volatility. Recent research efforts in the domain of solid-state electronics have proposed a new component named *Memristor* which has highly promising characteristics for enhancing or even replacing transistor-based components in current computer architectures [4]. In particular, in the domain of computer networks, its properties have a tremendous potential to enhance the energy efficiency, but also the computational capabilities of network packet processors [5]. A memristor compared to a transistor is characterized by state transitions which are non-volatile. In addition, such transitions do not support only read and write operations like in volatile transistor-based memory, but can also model so called cognitive computations as part of a state-transition. This way data storage and computations are co-located and make energy and time consuming data movements feasible. Although there is a rapid and tremendous interest in memristors, their implementation for network functions is challenging due to the absence of cognitive models benefiting from the memristive features of non-volatility and state-based response.

The implementation of memristors for cognitive packet processors poses several research questions in the areas of designs, implementations and tradeoffs in comparison to transistor-based packet processors. The non-volatile and state-based response of memristors encourages the exploration of its applications in cognitive systems, built upon the installation of programming rules for execution of network operations, called *rule-based systems*, e.g., Ternary Content Addressable Memory (TCAM) architecture for feeding and fetching cognitive rules. This leads to the fundamental research question, ''Can future computer networks be designed using memristor-based network components to benefit from the non-volatile state-based response and implementation of energy efficient network functions?''. Moreover, colocalization of memory and computation in memristors promotes its applications in novel cognitive architectures, which can analyze, develop and modify the rules based upon the system conditions, called *learning systems*, e.g., reconfigurable architectures, reservoir computation architectures, neural network architectures and neuromorphic computing architectures. Unlike in the von Neumann architecture, in a memristor-based architecture memory and computation is inherently colocalized. Hence, the next related research question is, ''In how far can memristors efficiently support cognitive computations with self-learning capabilities?''. The implementation of memristor-based architectures also comes at the cost of increased computational complexity and device implementation challenges. This motivates us for the next research question, ''What are the designs, trade-offs and challenges in shifting from the traditional transistor-based network components to the memristor-based network components?''. This survey focuses on comprehending these research questions related to memristor-based network components for the implementation of cognitive functions at packet processors in the Internet.

A. CONTRIBUTIONS

In this article, we survey and analyze research findings on memristive materials from the perspective of packet processors. In particular, we study recent advancements in memristor-based cognitive components that could be an integral part of future packet processing lines. In this context, firstly, this survey presents the design of rule-based systems using memristor-based TCAM architectures for incorporation of cognitive functions inside packet processors. It also presents the traditional measures of performance enhancement in TCAM with a comparison to the energy consumption statistics. Secondly, this survey studies the implementations of memristor-based learning systems for the development of self-learning cognitive functions at packet processors. This line of study includes reconfigurable architectures (including crossbar arrays, Field Programmable Gate Arrays (FPGAs) and application-specific architectures), reservoir computation architectures, neural network architectures and neuromorphic computing architectures. Lastly, this survey presents the design considerations and future directions of research for incorporation of memristor-based cognitive network architectures at packet processors. State-of-the-art literature has surveyed researches for designing memristor-based architectures, however, previous surveys mainly focused on memristive cognitive properties and specific architectures with less emphasis on the application-side of memristive cognitive architectures. On the contrary, our survey specifically focuses on the use of memristive components in packet processors to harness energy efficiency and cognitive functionality of memristors. However, the findings of previous surveys and reviews act as a foundation for our survey.

B. SURVEY FINDINGS

Our survey suggests that the memristor-based components can provide better performance in terms of energy efficiency and implementation of cognitive functions at packet processors than the traditional transistor-based components. In the rule-based cognitive systems, traditional transistor-based TCAM designs can only be slightly optimized in performance by managing hardware resources more efficiently.

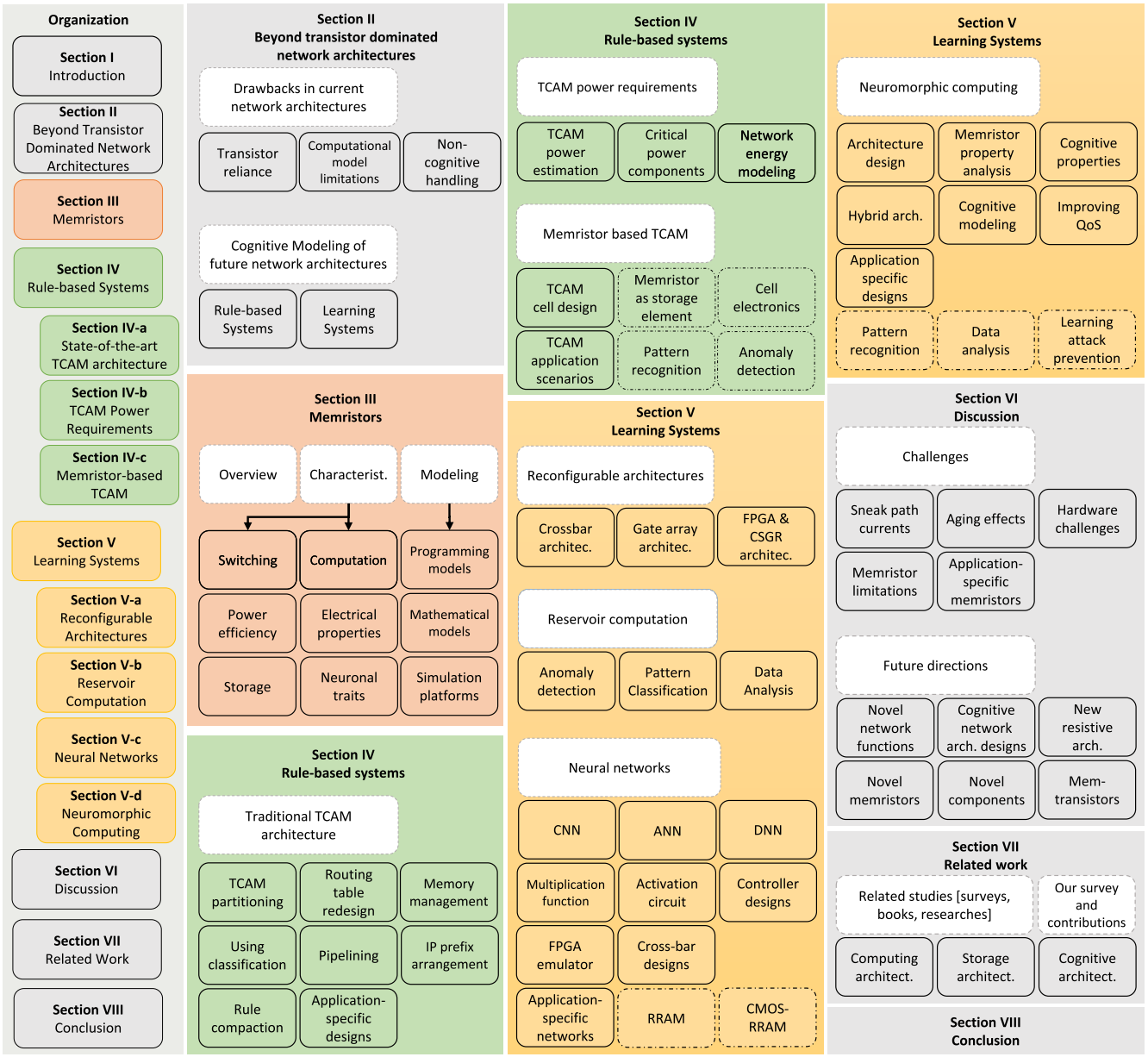

FIGURE 1. Organization of our survey paper.

On the contrary, memristor-based TCAM architectures can provide multiple programmable states, along with non-volatility, which support in developing energy efficient TCAM architectures for matching cognitive rules. In the learning systems, our survey shows that memristors can be used for the development of reconfigurable architectures like crossbar arrays, FPGA designs, etc. Memristor-based crossbar array designs provided sophisticated network functions stemming from matrix multiplication operations and development of cognitive functions. Moreover, the scope of network functions can be extended to incorporate cognitive packet handling by the development of memristive reservoir computation architectures. Our survey also showed that memristors can be used for the development of neural network architectures by programming *conductance* of memristors analogous to the programming of *weight* in neural networks. Lastly, our survey showed that memristors provide all the brain inspired synaptic properties which can make them an efficient alternate to transistor-based designs and they can be used in packet processors for providing cognitive capability. Based upon the discussed literature, our survey shows that the future computer networks can harness from memristor-based cognitive packet processors in terms of energy efficiency and cognitive functionality at packet processors.

C. PAPER ORGANIZATION

Organization of our survey paper in form of major modules and submodules is shown in Fig. 1. Section-II presents the

TABLE 1. Summary of important acronyms used in the survey paper.

Acronym	Definition
AC	Analog Current
ADC	Analog-to-Digital Converter
AI	Artificial Intelligence
ALU	Arithmetic-Logic Unit
ANN	Artificial Neural Network
ASIC	Application-Specific Integrated Circuit
BGP	Border Gateway Protocol
BNG	Border Network Gateways
CAM	Content Addressable Memory
CMOS	Complementary Metal Oxide Semiconductor
CSGR	Coarse Grained Reconfigurable Architecture
CNN	Convolutional Neural Network
DAC	Digital-to-Analog Converter
DNN	Deep Neural Network
DPI	Deep Packet Inspection
DRAM	Dynamic Read Access Memory
FPGA	Field Programmable Gate Array
GOPS	Giga [billion] Operations Per Second
GPU	Graphics Processing Unit
HLPM	Hardware-based Longest Prefix Matching
HTM	Hierarchical Temporal Memory
IC	Integrated Circuit
IFC	Integrate-and-Fire Circuit
IoT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol Version-4
IPv6	Internet Protocol Version-6
mRAM	Memristor Random Access Memory
MLP	Multilayer Perceptron
MSE	Mean Square Error
MTL	Memristive Threshold Logic
NIC	Network Interface Card
NVM	Non-Volatile Memory
QoS	Quality of Service
RAM	Read Access Memory
RegEx	Regular Expression Matching
ReLU	Rectified Linear Unit
RLC	Resistor-Inductor-Capacitor
RNN	Recurrent Neural Network
RRAM	Resistive Read Access Memory
SNN	Spiking Neural Network
SPE	Sneak-Path Encryption
SRAM	Static Read Access Memory
STDP	Spike Timing Dependent Plasticity
TCAM	Ternary Content Addressable Memory
XOR	Exclusive OR

motivation of the research problem. Section-III introduces the background and properties of memristors. Rule-based systems comprising of transistor and memristor-based TCAM architectures have been discussed in Section-IV. Section-V analyzes the memristor-based learning systems for cognitive network architectures including reconfigurable architectures (Subsec-V-A), reservoir computation architectures (Subsec-V-B), neural network architectures (Subsec-V-C) and neuromorphic computing architectures (Subsec-V-D). Memristive challenges and future scope of research are discussed in Section-VI. Section-VII presents the related surveys and reviews in comparison to our survey and Section-VIII concludes the paper. A list of major acronyms used in the survey paper has been summarized in Table 1.

II. BEYOND TRANSISTOR DOMINATED NETWORK ARCHITECTURES

In this section we explain why current network technologies are strongly limited in energy efficiency and in their ability to execute cognitive functions. Furthermore, we introduce the cognitive models we will use in the survey to classify advancements in memristors. In-network processing requires cognitive processing of network flows to deal with more network traffic with intelligent decisions, high throughput and less delay, at packet processors with power efficient network functions [6], [7]. On the contrary, current packet processors consume huge amount of energy resources and lack cognitive functionality for network packet flows [8], [9].

The root of these problems stems from the underlying operating principles of the transistor-based technology used in the current packet processors. The major problems in transistor-based architectures are volatility, high power consumption, large switching time and lack of state-based response. The transistor's state-based response resembles a Moore machine where the output depends only upon the input without any relationship to the past state of the transistor [10]. As a result, transistor-based architectures are used in von Neumann architectures which use a combination of memory and computation because transistors cannot compute and store at the same time. Moreover, the state of a transistor is volatile and requires continuous power supply for state maintenance. Also, power consumption in a transistor is the product of voltage and current across a transistor and it is solely dependent upon the hardware material characteristics. At the moment, transistor-based architectures consume large amount of power along with the generation of significant amount of heat due to the inherent material characteristics and atomic operations [11]. Regarding switching times, it is the time difference between an applied input and an observed output in a transistor, and it directly influences the available clock cycles per unit time for any network function. Transistor-based architectures are experiencing large switching times which worstly affects the computational performance of any operation. Moreover, invalidity of Moore's law in increasing the number of transistors per unit area and fabrication limitations exacerbated the exhaustion of resources for providing cognitive functions at packet processors. As a result of all these shortcomings, the implementation of cognitive functions for complex network operations, like network congestion management, dependent upon the past state of the network become challenging and energy inefficient to model in the transistor-based packet processor. The network functions, like anomaly detection, keep a track of past states of the system and current von Neumann architecture uses separate state storage and processing. As a result, it poses challenges regarding excessive power consumption and providing flow guarantees for cognitive network operations. Large switching time in transistors is a major challenge for the implementation of complex network functions because every network function

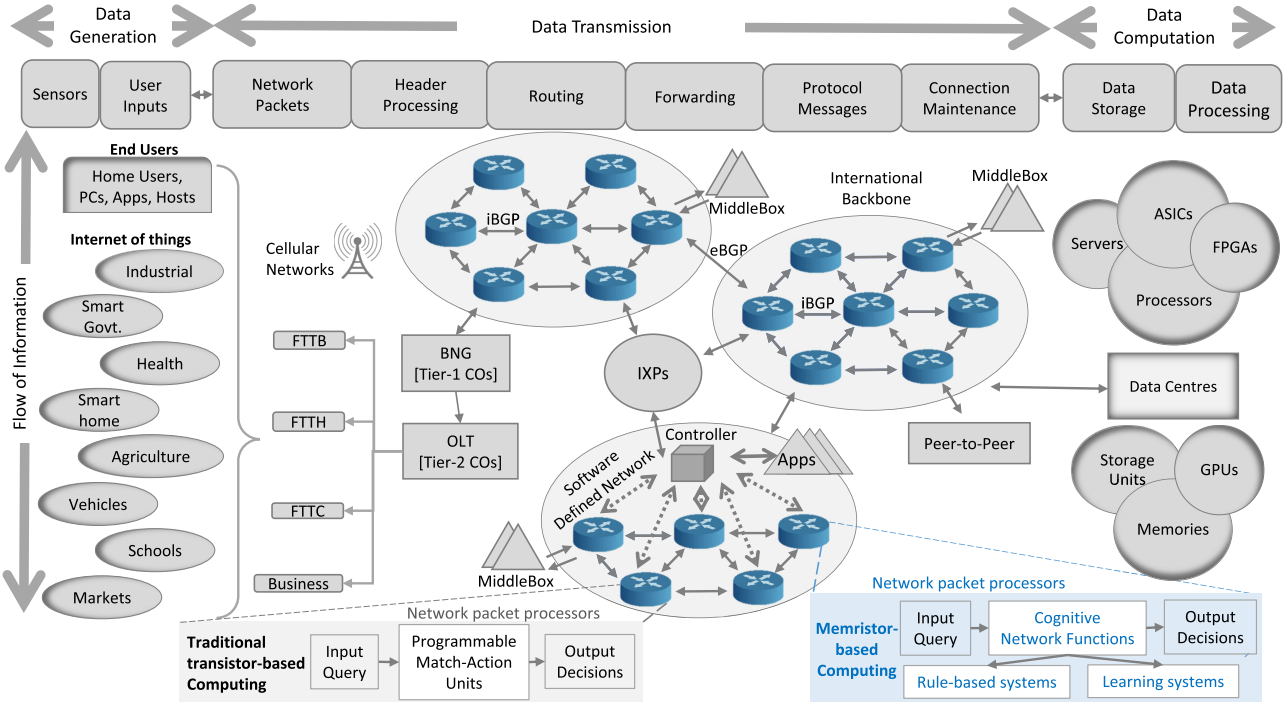

FIGURE 2. Taxonomy of the Internet architecture.

consumes more time for the same number of operations. Therefore, a fundamental question is whether and in how far alternative designs of packet processing platforms can help to counteract or even overcome all these limitations.

The current Internet architecture transports network packets between three major network entities; Data producers (end users and Internet of Things (IoT) devices), Data processors (communication system) and Data consumers (Data centers), as shown in Fig. 2. Data is generated by a range of end devices including IoT devices and end users amounting to 4.8 zettabytes of traffic in 2022 [12]. The *Data Transmission* phase comprises of a range of network appliances consisting of cellular networks and copper/fiber networks connecting to the Border Network Gateway (BNG). BNGs connect to the autonomous systems comprising of packet processors, with or without Software defined networks, and transmit the information to the *Data Computational* units, i.e., Data centers and remote servers, for processing and computing the packet payload information and replying Internet Protocol (IP) packets with the requested information. As evident from the Fig. 2, Data transmission phase carries the most load of traffic transmission and it's widely spread in the world and at the moment it is dominated by the traditional von Neumann architectural configurations built upon transistors. It takes a maximum of 40 Hops (15 hops for Routing Information Protocol [13]) for network packets to reach any connected device in the world and every hop count at a packet processor is associated with a delay for taking decisions based upon incoming packet header and performing non-cognitive computations. Implementation of cognitive functions at packet processors has already shown superior performance in decision making by incorporating a vast variety of rules without excessive consumption of storage space for incoming packets [14]. However, these cognitive benefits are reaped at the costs of increased energy consumption and excessive delays by using the transistor-based von Neumann architecture. The future packet processors can incorporate cognitive processing through incorporation of two kinds of systems in the edge and middleware nodes; (1) Rule-based Systems [15], (2) Learning Systems [16]. These two kinds of cognitive systems will be used to study advancements in the development of novel architectures at packet processors.

A. RULE-BASED SYSTEMS

Representation of knowledge is done in form of rules in a rule-based system. The programming model of a rule-based system comprises of *if-then-else* statements. There is no self learning capability in the system, but, the system is able to make cognitive decisions based upon a series of installed rules in the system. An example of rule-based systems in networks is the IP-lookup functionality at packet processors. The header information containing the destination IP address, VLAN ID, ethernet protocols, etc., of incoming packets is assessed to compute the destination path. This computation takes place by cognitive rules installation at specialized *Match-Action* units at packet processors as discussed below.

1) TCAM ARCHITECTURES

The high performance header processing of packet processors stems from the use of a specialized unit called TCAM [17].

TCAMs enable prefix matching in one clock cycle and are therefore one of the key building blocks of high performance packet processing in the Internet. Building on new data plane programming models, like the P4 language [18], packet processors can also be used for the acceleration of cognitive network functions by providing reconfigurability of processing pipeline at the hardware level with applications in content-based routing and load balancing. Nevertheless, a drawback is the high energy footprint, cost of the memory components, scalability issues and switching failures (e.g., Microsoft Azure switch failures [19]) of the packet processor.

B. LEARNING SYSTEMS

Artificial Intelligence (AI) empowered learning systems have a complex task of using the feature selection, analysis and classification to learn the rules with adaptive intelligence based upon the network function requirements without any influence from the outside factors. Learning systems can adapt to the knowledge and relearn and modify the rules for the network based upon the varying conditions. Network functions like firewall development, congestion control, load balancing, etc., can be hosted at packet processors to analyze the network packet streams and adapt the cognitive decisions based upon the network conditions. The implementation of learning systems can be performed by utilizing a diverse set of architectures, (1) Reconfigurable architectures, (2) Reservoir computation architectures, (3) Neural network architectures, and (4) Neuromorphic computing architectures, as discussed below.

1) RECONFIGURABLE ARCHITECTURES

Flexibility of the software can be used for high computing frameworks by employing reconfigurable hardware architectures. Programmable crossbar arrays and FPGAs are some of the examples of reconfigurable hardwares which can be used to implement cognitive functions. In crossbar array architectures, a set of inputs can be mapped to a set of outputs by connecting programmable entities (components) in between every input and output. FPGAs consist of programmable logic gates to map an input to an output and reprogrammability provides the ability to learn from the current inputs and modify the input-output logic by controlling the gate connectivity. Transistor-based technology consumes excessive energy resources and requires excessive data transport between memory and computational units which makes it infeasible to use in cognitive reconfigurable architectures. At packet processors, reconfigurable architectures provide the benefit of diversity in network functions for incoming packet streams. A variety of cognitive decisions can be installed and adapted based upon the network conditions.

2) RESERVOIR COMPUTATION ARCHITECTURES

In reservoir computation, a fixed and programmable nonlinear system called *reservoir* can be trained to map input signals to higher dimensional computational spaces. Reservoir computation uses the model of a recurrent neural network with feedbacks to map input and output signals. Every reservoir must have nonlinear functions for neurons along with information storage capability. However, transistor-based architectures require separate storage and computation to achieve this task which worstly effects the performance of the reservoir. In the field of networks, the network functions like congestion control, routing, load balancing etc. can be computed at packet processors by feeding parameters on the input side of the reservoir and complex cognitive decisions can be obtained on the output of the reservoir. The model can be trained to determine the optimal cognitive decision by programming the weights in the reservoir.

3) NEURAL NETWORK ARCHITECTURES

Cognitive decisions can be made through an Artificial Neural Network (ANN) which is made up of a network containing a circuit of neurons. The connections between neurons are modeled as programmable weights which can be tuned to achieve the desired performance of the network. Every neuron combines the inputs by a weighted sum and adds a bias after setting an activation function which controls the amplitude of the output. In the context of networks, neural networks can increase the range of decisions for network functions, like data analytics, at packet processors and they can transform the network from a rule-based network to a continuously evolving decision making network. However, transistor-based networks face limitations in neural network implementations due to network state requirements which requires continuous data movement between memory and computational units.

4) NEUROMORPHIC ARCHITECTURES

Significant amount of energy and time is invested in data transport between computation and memory units in the traditional von Neumann architecture [20]. To counter von Neumann issues, the neuromorphic architecture focuses on combining large number of computational nodes through an asynchronous clock. Moreover, all nodes have co-localized computational and memory units and every node takes a decision based upon the Spiking neural network model and propagates the decision through the network. Motivation of the neuromorphic research is derived from the functioning of human brain which operates at 1 exaflops (10^{18}) calculations per second with a 25 W energy consumption, while a supercomputer operates at 200 petaflops and consumes 10 MW energy for equivalent number of operations [21], [22]. Neuromorphic chips *Loihi* by Intel [23] and *TrueNorth* by IBM [24] are some prominent neuromorphic computing architectures. The packet processors can benefit from the neuromorphic architectures by deploying energy efficient network functions, like cognitive decision making using packet header fields and network conditions.

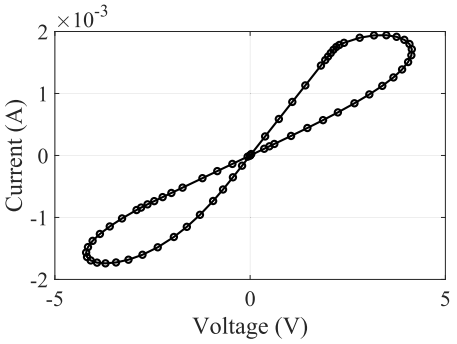

FIGURE 3. Pinched hysteresis curve of memristors.

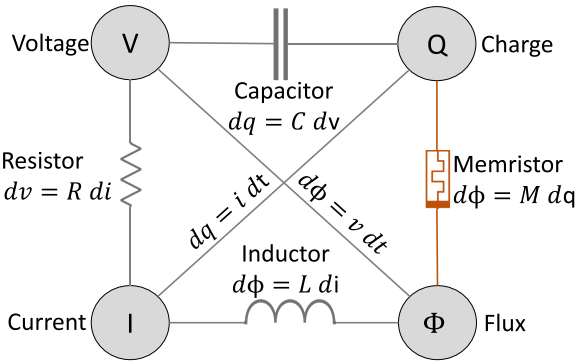

FIGURE 4. Fundamentals of electrical components in networks.

III. MEMRISTORS

In this section, we introduce the memristor, a fundamental circuit design component along with its background, history, characteristics, and modeling dynamics in reference to network packet processors. The Internet landscape is dominated by the transistor-based components used in computing and storage elements e.g., processors and network memories. However, continuous power supply is required to maintain the data inside the transistor-based components. Moreover, there is less diversity in the state-to-state transitions in the transistors which ultimately results in the scarcity of implementable functions or high overhead of implementation. As a result, the implementation of network functions built upon AI would require huge energy and excessive delays for data movement and huge costs would incur for state maintenance in the traditional transistor-based components. The remedy for all current network issues is the shift towards network entities built upon a novel component *Memristor* instead of transistors.

A. BACKGROUND ON THE WORKING OF MEMRISTOR

In a memristor, the state is represented by its electrical property, resistance (or conductance, the inverse of resistance). State changes require an applied voltage, but the state remains if no voltage is applied. To find the current state of a memristor, current flow is measured across the memristor. Analysis showed that the resistance exhibited by the memristor by applying an input voltage is a function of the past resistance, unlike transistor or any fundamental component like resistor, capacitor or inductor. This state dependent relationship of resistance can be represented in form of a pinched-hysteresis curve as shown in Fig. 3 (simulated in Matlab). Each point in a curve corresponds to a distinct state and number of states depend upon the composition and type of the memristor. In network analogy, it suggests that state of the memristor can be programmed and the memristive state is dependent upon the input parameters and past state of the memristor function. In terms of network functions, it gives the memristors an inherent ability for implementation of state-based functions including computational and memory operations because output is dependent upon the input and past state of the system and nonlinearity can be used for complex operations. It is also worth noting that the extreme right and left points are called *SET* and *RESET* states. These two points are used to eliminate the information about the past states, because, all past states converge to these two points by application of the required voltage. This response can be useful in removing the state-based memory in memristors for providing security and privacy in networking applications.

B. HISTORY OF MEMRISTORS

The concept of memristor has a short history; before 1971, the four fundamental electrical properties namely voltage, current, charge and flux, were linked together through three fundamental electrical components namely resistor, capacitor and inductor. In 1971, Leon Chua proposed a new theoretical fundamental component namely *Memristor* by linking the four fundamental properties together [25]. Fig. 4 presents the linkage between the four fundamental electrical components by voltage, current, charge and flux. Memristor was introduced as a theoretical linkage between charge and flux. Memristor was proposed to be the fourth critical building block in circuits along-with resistor, capacitor and inductor [26]. In a resistor, induced current is dependent upon the applied voltage only. On the contrary, current in a memristor is dependent upon the past state of the memristor as well as the applied voltage. This gives it the name *memristor* from the combination of *memory* and *resistor*. Memristor behaves similar to a nonlinear resistor with a memory, and such a behavior cannot be modeled by any combination of a Resistor-Inductor-Capacitor (RLC) circuit because all previous circuits are using stateless operations. In 2008, Stanley Williams and his team [4] discovered the fourth fundamental element *Memristor* at a research lab. They showed that memristance arises naturally for nanoscale electronics by coupling together the basic physical properties of ionic transport and solid state electronic after application of an external voltage [27]. The relationship between voltage and current was demonstrated by varying the width of the doped region. Layers of Titanium oxide and oxygen deficient Titanium oxide sandwiched between Platinum electrodes were used for the fabrication of the first memristor, as shown

in Fig. 5. Based upon the feature study, memristor discovery proved to be a milestone in developing non-volatile and state-based network components which can ultimately lead to cognitive computational and memory architectures [28].

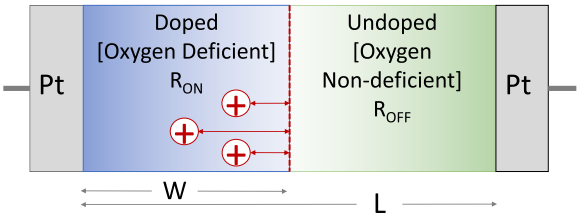

FIGURE 5. Structure and composition of a memristor.

C. CHARACTERISTICS OF MEMRISTORS

In the context of networking architectures, we are interested to understand how the memristors would impact the performance and energy-efficiency of packet processors as compared to the traditional transistor-based components. Therefore, we survey the research focusing on the analysis of hardware characteristics of memristors critical for packet processors. We focus on memristive switching behaviors, combinational characteristics with other components, cognitive properties, electrical properties for read and write operations, and power efficiency traits of memristors in this section.

1) SWITCHING BEHAVIOR

Memristor switching refers to the transition of a memristor in between maximum and minimum resistance states in a binary state transition model. Memristors have unique properties for switching behaviors in packet processors. Prodromakis et al. [29] studied the switching mechanism of memristors at nanoscale level. The authors performed experiments on a four layer fabricated memristive device and presented the current-voltage and memristance (electrical resistance of memristor) patterns in comparison to the hysteresis curves. By observing the stable and continuous state transitions in memristors, the study concluded that memristors can be a promising component for switching mechanisms with better energy efficiency and switching performance. Moreover, the research showed that memristors are feasible at nanoscale for the requirements of practical applications like computational and storage architectures in the networks.

2) COMBINATIONAL SWITCHING CHARACTERISTICS

Owing to the widespread use of network functions built upon resistors, capacitors and inductors, it is highly essential to study the memristor properties in combination with these components. Joglekar and Wolf [30] showed that memristor's hardware characteristics in form of current voltage patterns and complex logic development can be enhanced by combination with capacitors and inductors. The major gains in combination with other components were achieved in form of variations in hysteresis curves, and decaying and damping properties of circuits. All of these parameters are linked to the state and state transitions of memristors which are the basic building blocks of network functions. In network analogy, it suggests that the combination with other components can help us in varying the state dynamics of memristor-based architectures by controlling the input parameters and implementing a variable output function.

3) COGNITIVE PROPERTIES

Cognitive capability can be employed in networks using memristors by comparing the brain's neuronal properties with memristors. Sheridan et al. conducted a survey-based study and showed that the critical features of human brain including Spike Timing Dependent Plasticity (STDP), short term plasticity and long term potentiation can be exhibited by memristor-based networks [31], [32]. All of these features relate to the strength, connectivity and output behavior of the brain's neural network. The study shows that the hardware topology of memristor poses several challenges including *Sneak path problem* which refers to unwanted current paths and creates noise in the system. However, combination of Complementary Metal Oxide Semiconductor (CMOS) and memristor-based components can be advantageous for achieving the functionality required in network functions. Authors showed the physical characteristics of memristors through modeling and motivated the use of cognitive components in future computer networks.

4) THREE TERMINAL CONTROLLABLE CHARACTERISTICS

A critical line of research is the study and development of three terminal memristive components, instead of the typical two terminal memristors. A third terminal can control the output and it can be employed to use the memristor as a switch. Duan et al. present the switching dynamics of oxide-based memristors and study the characteristics of three terminal memristors [33]. The authors present a novel three terminal device for emulating the heterosynaptic plasticity and compare it with HfO_2 based memristors to show the ability of mapping neuronal functions. The study suggests that a fuzzy restricted Boltzmann machine can bypass the inherent shortcomings of device variations and resolve device stochasticity in neuromorphic circuits. Moreover, the proposed device can implement the non-Boolean logic which suggests its effectiveness for employment in neuromorphic applications.

5) ELECTRICAL PROPERTIES

The analysis of the structural properties and electrical characteristics can aid in determining the read and write operations of the memristors. In write operation, a certain voltage, depending upon the material characteristics, is applied to bring the memristor to the required state. On the contrary, read operation refers to the application of small pulses of voltage in order to measure the current to retrieve the state of the memristor. Ho et al. [34] present the electrical properties of

memristors in relationship to the read and write operations. The authors develop closed form expressions to emphasize the practicality of memristor-based read and write operations in a crossbar array architecture. In a follow up work [35], the authors show that data integrity and noise tolerance are major issues in nanoscale packing of memristor-based architectures. The closed form mathematical derivations showed the practicality of memristors as data storage devices.

6) POWER EFFICIENCY TRAITS

Understanding the energy characteristics of hardware materials plays a pivotal role in the practical realizations of novel components. Radwan and Fauda [36] made a thorough investigation on the mathematical modeling of memristors along with the estimation of energy consumption and associated combinational factors. The authors presented the memristive circuits and developed the Arithmetic-Logic Unit (ALU) for memristor operations. The review concluded that memristors can be utilized in combination with transistors for realization of a range of circuits, and Spice simulation models can be used to estimate the performance of hardware designs. Moreover, the review showed that memristors are power efficient components due to their state retaining capabilities.

7) SUMMARY OF MEMRISTIVE CHARACTERISTICS

In this section, we reviewed the researches related to the fundamental properties and characteristics of the memristors which can aid in network-based memristive applications. Review showed that memristors can exhibit stable and continuous state transitions in form of timing delays and switching characteristics. Combination with resistor, capacitor and inductor can help in achieving stable state transitions based upon the developed networking algorithm. This switching behavior can aid in memristor-based novel architectures and it can be utilized to replace the state-of-the-art transistor-based architectures with state-based components. Recent researches on three terminal memristors also showed performance benchmarks for implementation in complex computational architectures. By using three terminals, a control logic in networks can be established using memristors and this makes the memristor useful both as a storage and computational element. Hardware characteristics of memristors also showed brain like neuronal traits including STDP and potentiation factors. It suggests that memristors can be utilized to map complex cognitive neural functions in packet processors. Memristors exhibit prominent electrical properties, like power efficiency in read and write operations, with several challenges, including optimal circuit designs, open for future research. All of these factors promote the use of memristor as a promising networking component for the design of future computer networks.

D. MEMRISTIVE MODELING

Memristor-based researches have spanned over the development of computation and storage components, and it is highly important to model the behavior of a memristor in a simulation and analytical framework for future cognitive packet processors. In this section, we would highlight the researches which modeled the behavior of a memristor. Fig. 6 presents an overview of the modeling domains catered in the related literature regarding memristors.

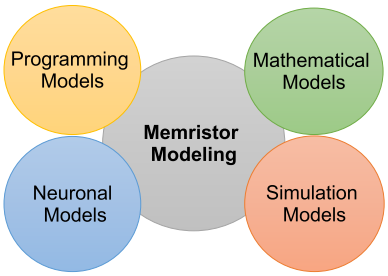

FIGURE 6. Dimensions of memristor modeling in various domains.

1) MEMRISTOR PROGRAMMING MODEL

Bala et al. [37] made the first effort for modeling of memristor in *C++* programming language. Authors focused on the development of a memristor-based single and multi-layer neural network in *C++*. Simulation results for pattern classification at two learning rates for linear and nonlinear functions demonstrated that memristor can perform the role of classification with significant accuracy by using the proposed model.

2) MEMRISTOR-BASED NEURONAL MODEL

In another study, Zhevnenko et al. [38] simulated a memristor-based model for emulating the spiking response similar to a neuron. Researchers incorporated the approximation of memristor switching series and mobility functions, and applied it to an experimental data set of ZrO_2 based memristive device containing 2000 switching cycles. The study showed that memristors can significantly reduce power consumption and achieve better performance in simulating a neuromorphic architecture.

3) MULTI-STABLE MEMRISTOR MATHEMATICAL MODEL

Apart from simulations, Lin et al. [39] proposed a mathematical model for a memristor with multi-stability. The research showed that a multi-stable memristor can demonstrate many co-existing pinched hysteresis loops under varying states. Application of the proposed model in a Hopfield neural network demonstrated many coexisting chaotic attractors which can increase the ability and performance of a neural network.

4) MEMRISTOR SIMULATION PLATFORM

Xia et al. [40] proposed a simulation platform for memristor-based neuromorphic architectures called *MNSIM*. A hierarchical structure was developed for flexibility and integration with other components. Moreover, a computational accuracy model was developed and incorporated in MNSIM in order to analyze the non-ideal device factors and interconnection

challenges. The proposed MNSIM proved to be 7000 times performance efficient than the state-of-the-art SPICE model and it provided optimality in design decisions.

5) SUMMARY OF MEMRISTIVE MODELS

A review of memristor modeling showed that the modeling of memristors is feasible and realizable, and theoretical results proved the simulation results in the previous studies. Current researches have developed the memristor models from programming languages, like *C++*, to the development of independent simulation platforms. Researchers also realized the feasibility of memristor's mathematical and neuronal models. State-of-the-art literature showed that memristors can be modeled along with efficient characterization of all required network modeling parameters.

IV. RULE-BASED SYSTEMS

Current packet processors are a combination of computation and storage units which are used to implement cognitive functions. These cognitive functions can implement intelligent operations in networking applications like routing, firewall, load balancing, etc. in any network using rule-based systems. In these systems, network memories play a crucial role for operation of network functions in storing the rules and performing matches against packet header fields in every clock cycle. Among the rule-based network components, TCAM is the most critical resource due to high energy consumption, complex functionality of performing match operations within a single clock cycle, limited memory size and high cost. In this section, we firstly survey the state-of-the-art research and performance improvement techniques in TCAM architectures. Later, we study the power requirements of TCAM and critical power hungry components. Lastly, we survey the memristor-based TCAM researches including design goals and performance analysis. Fig. 7 shows the pie chart for classification of TCAM researches. Analysis shows that TCAM performance analysis has been the most prominent research direction followed by the application of memristors in TCAM.

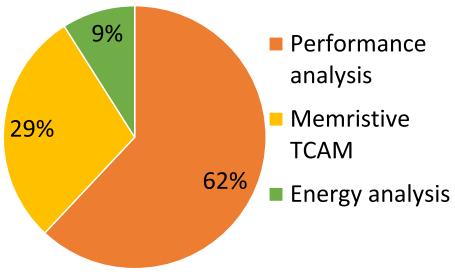

FIGURE 7. Classification of TCAM researches.

A. STATE-OF-THE-ART TCAM ARCHITECTURE

TCAM is the most crucial network resource which performs the matching and comparison operations in any network component within one clock cycle. Before delving into the details of future memristive architectures for TCAM, some of the preceding researches and improvements for the TCAM are presented in this subsection. Fig. 8 presents an overview of major research domains regarding performance improvements in the traditional TCAM architecture.

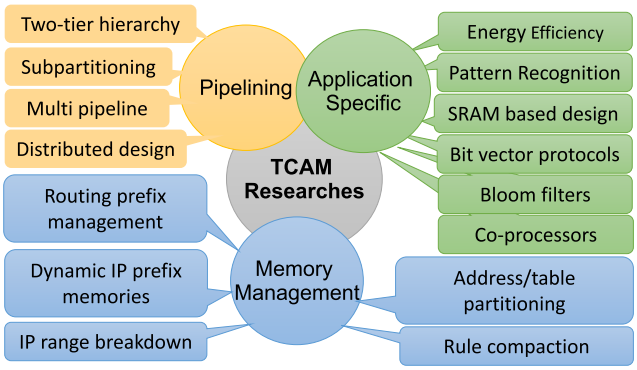

FIGURE 8. Taxonomy of TCAM state-of-the-art researches.

1) ENERGY EFFICIENT HIERARCHICAL TCAM

Pattern classification in TCAM can be performed with less energy consumption by using a hierarchical TCAM architecture. Spitznagel et al. [41] suggested a two level hierarchy for the TCAM architecture where an index block can be enabled/disabled for querying the main block and the secondary block can be used for performing the circuit-based comparison operations. The proposed TCAM design can perform 100 million lookups per sec for 100, 000 filters. Moreover, power and space efficiency are achieved by a reduction factor of ten and three times, respectively. However, the issue of incremental updates and more elaborate research on block size and improvements in hierarchical structure are required in future research.

2) HIERARCHICAL TCAM FOR ROUTING TABLE COMPACTION

High power consumption and heat dissipation in TCAMs have motivated the researchers to look into the options of routing table compaction. Ravikumar and Mahapatra [42] performed research on minimizing the memory consumption along with the feature of IP route increments in TCAM architectures. The authors proposed a two level hierarchical architecture in which the former stage deals with the compaction of required memory and later stage deals with the selective selection of only a subset of the TCAM. Prefix compaction, prefix aggregation and routing table compaction were the major design parameters in the proposed research. Simulations and analysis on a router data set, containing prefix addresses, showed that the power savings can range from 67% to 99% due to compaction and other features. However, the scope of this study limited to the application side of TCAM architecture.

3) TWO-TIER PREFIX MATCHING STRATEGY

Research on novel hardware-based lookup for longest prefix matching in pipelined TCAMs can also improve the energy efficiency of TCAM. Kasnavi et al. [43] suggested a two tier approach for Hardware-based Longest Prefix Matching (HLPM). First stage focuses on the search of a *don't care* bit in the prefix ending for a partial match. A simplified cell design has been proposed to reduce the complexity and energy consumption in this stage. In the second stage, complete output (full match) is forwarded based upon the *max* value of coded lengths of prefixes. The proposed technique demonstrated a 30% reduction in power consumption, along with less area consumption.

4) MULTI-PIPELINE PARTITIONING ARCHITECTURE

The requirement of multiple clock cycles in search operation for multiple queries can ultimately lead to excessive energy dissipation in high work load scenarios. To tackle this issue, Jiang and Prasana [44] proposed a partitioning based multi-pipeline architecture. Authors proposed the use of *HyperCuts* packet classification algorithm followed by a decision tree with bounded height. Two different techniques were employed to partition the decision tree into several disjoint sub trees and map it into Static Read Access Memory (SRAM)-based pipelines. The study showed that the proposed SRAM-based architecture can store 10 K rules in 0.336 MB with 8 pipelines and provide 1 Tbps throughput with a 2.25 times reduction in energy consumption. In another study, Jiang et al. [45] enhance the same architecture to develop memory distribution and management across multiple pipelines. Moreover, strategies to load balance the traffic across multiple pipelines are also proposed. Crux of the idea lies in the sub-partitioning in form of trie and sub-trie units in order to attain an efficient memory balancing architecture. The proposed strategy showed to be highly efficient by storing 200 K unique router prefixes using only 3.5 MB cache. Moreover, throughput up to 1 Tbps was achieved with 10 byte packet sizes. However, implications and implementations in network switches need to be explored further for practical implementations of the proposed technique.

5) DISTRIBUTED TCAM ARCHITECTURE

Study of the distributed architecture in TCAM is a promising avenue of research for improving the TCAM performance. Zheng et al. [46] proposed a distributed TCAM architecture for ultra-high lookup throughput and efficient memory management. The authors proposed the use of multiple TCAM chips utilizing load balancing along with an improved performance algorithm for the mapping of TCAM table entries. The study focuses on the idea of decreasing the number of lookup operations by executing search operation inside a single sub-TCAM chip only. The results showed that 25% more TCAM entries can be accommodated besides providing throughput of 533 MHz by employing four 133 MHz TCAM chips. The only drawback of this line of research is the incorporation of mapping overhead for the arrived entries inside the TCAM table.

6) EFFICIENT MEMORY MANAGEMENT

Performance and energy efficiency of TCAM can also be enhanced by use of efficient memory management mechanisms. Mishra and Sahni [47] proposed a dual TCAM architecture (called *DUOS*) for management of routing tables in TCAM. The authors proposed four memory management schemes which revolve around the idea of supporting control plane incremental updates without delaying the lookups in the data plane. The proposed strategy showed 6-7 times less power consumption than the competitor technologies. However, the scope of this research was limited to memory management and significant research is required to explore its applicability for the networking scenarios requiring sophisticated network functions.

7) OPTIMAL ROUTING PREFIX

Management of the routing table is a critical step in increasing the performance of TCAM. In this regard, Mishra and Sahni [48] studied the routing table and suggested the use of optimal routing prefix for increasing the performance of TCAM. Although, the proposed design called *PETCAM* takes only one clock cycle for search operation, but, it requires additional resources in the follow-up step of SRAM evaluation which makes it power and energy hungry. To overcome this limitation, the authors proposed improvements in SRAM-based architecture in order to avoid energy wastage and performance degradation. Experimental results demonstrated that the proposed TCAM scheme can reduce power consumption and memory requirements from 8-98% and 45-78%, respectively, based upon the network scenarios.

8) DYNAMIC ROUTE PREFIX MEMORIES INSIDE TCAM

Redesigning of routing table can also aid the network in improving performance benchmarks in a TCAM memory [49]. Akhbarizadeh et al. [50] exploit the inherent characteristics of the Internet by employing small scale memories which contain the popular route prefixes inside the Application-Specific Integrated Circuits (ASICs). Moreover, partitioning of the TCAM chip into submodules and taking advantage of the dynamic memories increased the throughput by more than six times with less power consumption as compared to the traditional design. A major advantage of the proposed strategy is the availability of multiple search operations per clock cycle in certain scenarios. The scope of this research needs to be explored and validated for complex TCAM pattern recognition scenarios like Internet Protocol Version-6 (IPv6)-based networks.

9) LOOK-UP USING IP-RANGE BREAKDOWN

Managing IP-based lookup operation requires critical analysis of the storage and update scenarios of IP prefixes.

Chang [51] proposed a two-level TCAM architecture focusing on the IP lookup scenarios for fast and efficient lookup. Author suggested the use of IP range breakdown in order to avoid range-to-prefix blowout scenarios, and proposed two range-to-prefix conversion schemes which can increase the performance and efficiency of TCAM. Depending upon the range type, only *4n-3* and *2n+1* TCAM storage entries are required for contiguous and non-contiguous ranges, respectively. Moreover, the proposed scheme incorporates the operations of insertion and deletion of new entries in the lookup tables. Experiments showed that the proposed architecture provides better performance in TCAM lookup time and memory requirement.

10) PARTITIONING PREFIX ADDRESSES

Search operation in prefix-based assignment in a TCAM table can be improved by a priori sophisticated partitioning of prefix addresses. Panigrahy and Sharma [52] suggested the partitioning of TCAM prefixes into eight groups, such that only a particular subgroup is searched during any operation. Significant power can be conserved by employing multiple sub TCAM chips and using only a selected sub TCAM chip at a time depending upon the prefix assignment. Under ideal circumstances, where traffic distributions and patterns are known apriori, the partitioning operation can provide up to 1000 million lookups per second. The major limitation of this scheme is its anomalous behavior for small enterprise networks where traffic characteristics vary with time. Also, in large enterprise networks, a significant overhead is expected for anomalous traffic.

11) ROUTING TABLE PARTITIONING

TCAM routing table refinement can also aid in minimizing the energy consumption of TCAM. Zane et al. [53] proposed two TCAM-based forwarding engine architectures based upon the bit selection and trie-based architecture configuration. The basic idea lies in the same argument that search operation should be performed in a limited subsection of TCAM memory only. Both bit selection and trie-based architecture provide the partitioning mechanisms with inherent pros and cons. Depending upon the upper bound of power and energy consumption, the authors chalked a layout for TCAM design decisions for the hardware design.

12) RANGE EXPANSION USING RULE COMPACTION

Range expansion is another critical problem for TCAM due to the growing number of classification rules in TCAM. To prevent range expansion problem, Liu et al. [54] proposed a novel design called *TCAM Razor* in order to incorporate vast number of rules in the limited size of TCAM memory. The multidimensional rule list is transformed into one dimensional list and decision-based diagrams are employed to generate equivalent compressed rules performing same operation. TCAM Razor showed a compression efficiency of 31.3% and 29%, respectively. An added advantage of this software-based improvement is that it does not require any hardware modifications in the current TCAM architecture. The scope of this study is limited to the objective of incorporating more rules without occupying additional space in the TCAM architecture.

13) SRAM-BASED ARCHITECTURE

TCAM energy and performance limitations can be bypassed by designing an SRAM-based architecture which is much better in conserving energy and minimizing space utilization. Ullah et al. [55] proposed a novel Z-TCAM architecture which employs the TCAM functionality with an SRAM-based architecture. Basic idea lies in the partitioning of TCAM columns and rows into hybrid TCAM sub-tables which can be mapped into distinct memory blocks. Two implementations of Z-TCAM on 512×36 and 64×32 sizes showed that the proposed design is realizable and technically feasible. A major drawback is the search latency in Z-TCAM which spans to three clock cycles, instead of one clock cycle in traditional TCAM.

14) BIT VECTOR PROTOCOL FOR PACKET CLASSIFICATION

Mapping and forwarding of packets through a large number of classification operations in TCAM, especially at line rate configurations, is a major challenge. In this regard, Baboescu and Varghese [56], [57] performed research on the proposition and refinement of a bit vector protocol algorithm for packet classification. The authors proposed improvements including recursive aggregation of bitmaps and filter rearrangement techniques in order to increase the performance. The results showed that the proposed technique gives better performance in achieving packet classification due to the incorporation of SRAM. The importance of this research lies in opening the horizon for future researches in the domain of application-specific hybrid architectures.

15) COMPLEX PATTERN RECOGNITION

Malicious attacks pose a serious threat to the Internet and they can be efficiently addressed through a TCAM-based pattern matching architecture. Yu et al. [58] showed that complex patterns including arbitrary long patterns, correlated patterns, and patterns with negations can be dealt effectively with a throughput rate of 2 Gbps with a 240 KB TCAM for a ClamAV data set of 1768 patterns. Owing to the wild card field of TCAM, it has an added ability of correlating patterns and detecting patterns with negations and wildcards. The fundamental motivation in using TCAM is to limit the security overhead on the client/server side.

16) QUERIES USING BLOOM FILTERS

The search and match operation can be revolutionized by employing bloom filters for resolving search queries. Dharmapurikar et al. [59], [60] proposed a novel concept of using bloom filters for longest prefix matching. The major advantage of this research dimension is the fact that performance becomes independent from the length of the prefix. This fact motivates its usage for lookup operations in

both Internet Protocol Version-4 (IPv4) and IPv6 addresses. Results spanning over IPv4 Border Gateway Protocol (BGP) tables showed that the search operation completes in 1-2 hash probes per lookup depending upon the data set. However, this strategy has no search guarantees (like TCAM) which makes the delays quite variable.

17) TCAM AS CO-PROCESSOR

Owing to the search capabilities of TCAM, there are vast applications which can benefit from not only search but also computational powers of TCAM. A major advantage of TCAM is the ability to perform efficient on-chip logic minimization. Ahmad and Mahapatra [61] presented the TCAM computational design with minimum required resources for using TCAM cells as a co-processor. The study showed that incremental insertion and bulk deletion can be achieved within 0.25 μs and 3.8 *ms*. Moreover, compaction of $10,000$ entries can be achieved in less than 25 *ms* by employing only 300 TCAM entries.

18) LESSONS LEARNED

The most prominent research direction in TCAM is the hierarchical or pipelining architecture for TCAM which focuses on splitting a TCAM module into submodules for energy conservation and comparison operations. This technique can save significant power and provide better performance in terms of number of search operations required for the look-up scheme in TCAM. Another major research domain is the memory management in TCAM which includes routing table management, prefix address optimization and sub-memories assignment for routing prefixes. This domain strengthened the incorporation of more rules with less memory inside the TCAM and increased the scope of TCAM for network switches with complex cognitive functions. The development of application-specific TCAM architectures is another avenue of research. TCAM can be used for computational processing, filtering techniques, classification tasks, SRAM-based performance enhancements and development of application-specific TCAM architectures. This interdisciplinary research domain overlaps with the domains of information processing, big data and computer architectures for performance improvement in order to increase the performance of TCAM. Nearly all the studies pointed out the shortcomings of current TCAM architectures including limitations in match table entries, limited number of rules, huge memory requirement, excessive energy consumption, high cost, scalability issues and lack of non-volatility characteristics. No research has been able to fulfill the demands of the current TCAM architecture in terms of all these requirements due to the hardware limitations of current components.

B. TCAM POWER REQUIREMENTS

Among all storage architectures including SRAM, Dynamic Read Access Memory (DRAM) and flash, TCAM consumes the most amount of power as shown in Table 2. In this subsection, we aim to survey the TCAM power requirements and energy footprint of the current network infrastructure.

TABLE 2. Power consumption in various network memories [62].

Technology	Power	Max size	Frequency
TCAM	3 μW	32 MB	360 MHz
SRAM	40 nW	200 MB	633 MHz
DRAM	250 pW	64 GB	1333 MHz
Flash	0.3 pW	2 TB	–

1) TCAM POWER MODEL

TCAM consumes huge amount of power and energy resources which also adds an indirect burden on the cooling budget of the network devices. Agrawal and Sherwood [63] estimated the power requirements among various components in a TCAM cell. The authors developed a power model for TCAM and verified it from the available industrial data sets. The study showed that majority of the TCAM power is consumed by the *match lines*, followed by the *search lines* and *priority encoders*, respectively. The research also showed the consequences of increasing the TCAM table size and it showed that increasing the number of columns in the TCAM table has less drastic effect on energy consumption than increasing the number of rows in the TCAM table. Analysis showed that the power consumed by the search operation of TCAM is comparable to a similar search operation in SRAM and it motivates the usage of hybrid SRAM and TCAM architectures in future networks.

a: ENERGY FOOTPRINT FOR FUTURE ARCHITECTURES

Power factors in future Internet architectures are quite critical for design of new devices. Chen et al. [62] studied the energy footprint and power efficiency for the future computer networks. The researchers developed router power model, data plane power consumption model and performed power comparison for various architectures. The results showed that the IP core and NDN core routers consumed less power than an IP edge, NEBULA edge, SCION edge and NDN edge routers. The study concluded that packet carried state is more power efficient than the routing table lookup. Moreover, end-to-end communication consumes less power than the use of in-network caching. Also, analysis showed that pervasive caching and edge caching have nearly identical energy footprints.

2) POWER EFFICIENT MATCH LINES

As suggested in previous researches, match lines consume majority of the power by continuously supplying voltage pulses. Mathan and Ravichandran [64] proposed a novel data aware AND-type match line architecture for TCAM. The authors designed a TCAM of 256 × 128 bits on 90 nm technology and showed that it provides an improvement in speed and power by 35% and 45%, respectively. The proposed strategy is dependent upon the inter-data and intra-data dependencies and the TCAM cell design contains a

Data Aware block which plays the most crucial role in the evaluation of data dependencies. The performance of this strategy is very much limited for independent and identically distributed data sets and the scheme is highly dependent upon the data analytics approach.

3) LESSONS LEARNED

In this subsection, we summarize the findings regarding TCAM power requirements. Preceding researches showed that TCAM is the most power hungry component among other memories including SRAM, DRAM, etc. The high energy footprint stems from the complex match operations performed inside every cell in the TCAM. Majority power in a TCAM architecture is consumed by the *Match Lines* which have to be recharged after every clock cycle. Power models are also available for TCAM which can be used to map, measure and compare the performance of TCAM in reference to other networking models. All of the studies conclude that energy efficiency is the most critical problem in the TCAM architecture and it motivates us to study memristor-based energy efficient designs for TCAM.

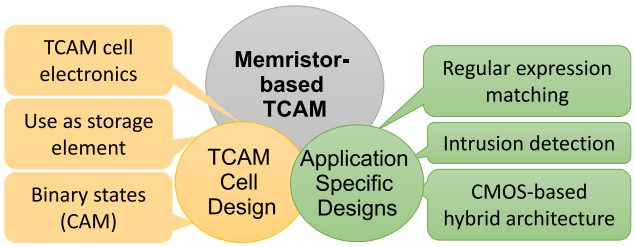

FIGURE 9. Current research hierarchy in the domain of memristor-based TCAM.

C. MEMRISTOR-BASED TCAM

Memristors can be employed in TCAM by using their non-volatility and cognitive properties. In this subsection, we survey the memristor-based TCAM researches. There are two lines of research in designing memristor-based TCAM; either designing a memristor-based TCAM cell or proposing application-specific TCAM architectures. We analyze the researches in both of these domains to study the practical implementations of memristors. Fig. 9 presents a glimpse of research dimensions in the memristor-based TCAM along with key focus areas. Some more recent research findings have just become available at the time of this publication.[1,2]

[1] S. Saleh, A. S. Goossens, T. Banerjee, and B. Koldehofe, "TCA*m*M$^{\text{CogniGron}}$: Energy Efficient Memristor-Based TCAM for Match-Action Processing," To Appear in *Proceedings of the 7th International Conference on Rebooting Computing (ICRC 2022)*, IEEE.

[2] S. Saleh, A. S. Goossens, T. Banerjee, and B. Koldehofe, "Towards Energy Efficient Memristor-based TCAM for Match-Action Processing," To Appear in *Proceedings of the 13th International Green and Sustainable Computing Conference (IGSC 2022)*, IEEE.

1) TCAM CELL DESIGN

TCAM cell design is the most crucial step in TCAM architecture formulation because it has to complete all search operations within one clock cycle. A number of studies (including [65] containing a test chip design) laid the foundation of initial research by proposing TCAM cells based upon transistors. However, the state-of-the-art designs have constraints in form of energy efficiency and development of complex network functions. The added benefits of memristors in terms of energy efficiency and memory characteristics have motivated various researchers to investigate the TCAM cells based upon memristors. In this subsection, we would present the TCAM cell design based upon memristors.

a: ELECTRONICS OF TCAM CELL

The nanoscale design of TCAM cell using memristors and transistors is a first step in the TCAM cell design. Guo et al. [66] proposed the electronics of a TCAM cell based upon memristors. Considering the huge power dissipation and off-chip bandwidth restrictions, the study showed that memristor-based TCAM cell can provide better computation power on the chip. The results showed that memristor-based TCAM cell improves performance and energy consumption by four and ten times, respectively. Moreover, memristor-based TCAM cells can provide twenty times better resource density than the CMOS-based cells. However, the scope of this study is limited to the hardware part of TCAM, and authors did not focus on the computer network protocols and packet processing techniques.

b: MEMRISTOR AS A STORAGE ELEMENT IN TCAM

Memristors are a promising storage element due to their non-volatile nature. Junsangsri and Lombardi [67], [68] proposed the use of two memristors inside a TCAM cell in series to perform the read, write and search operation in TCAM. Simulations over HSPICE showed that the memristor-based TCAM cell provides a write time in between 60-70 ns depending upon the voltage variations. Moreover, the query search time for memristor-based TCAM decreases by around 700 ps by changing fabrication technology from 65 nm to 32 nm. Extensive research in this study suggested that the memristor-based TCAM can provide better energy efficiency in less chip area with enhanced read/write operations.

c: MEMRISTOR-BASED CONTENT ADDRESSABLE MEMORY

Instead of the research on TCAM design, some initial studies focused over a binary Content Addressable Memory (CAM) working on two possible states only. In this regard, Chen et al. [69] propose a Memristor Content Addressable Memory (M-CAM) architecture by placement of memristors inside the CAM cell. The authors developed a current controlled mechanism for memristor-based CAM cells in order to perform the read and write operations in CAM. The proposed fuzzy look-up functionality for CAM is the

basic contribution of the researchers. However, scope of this research did not span over the domains of packet flow and traffic analysis in a fully developed CAM architecture.

d: DATA WRITE OPERATION IN TCAM

Date write operation is a challenging task in memristors due to the unwanted current paths and memristive state-based response. Ruotolo et al. [70] suggested that a memristor-based memory can be much more faster than a phase change memory and much simpler than a magnetic memory. To mitigate issues in write operation, researchers proposed a memristor-based write operation for the memory operations. Analysis of the proposed scheme, carried out at various voltages and memristive parameters, showed that the proposed write operation can cope fairly with noise and memristor-based memories can benefit from similar write operation strategies.

2) APPLICATION-SPECIFIC DESIGNS FOR MEMRISTOR-BASED TCAM

The design of memristor-based TCAM depends upon the developed applications and a review of these studies is presented in below lines.

a: REGULAR EXPRESSION MATCHING

Graves et al. [71], [72] conducted an indepth analysis on a memristor-based TCAM for Regular Expression Matching (RegEx). The authors showed that the incorporation of memristor increases the capability of TCAM in decreasing power consumption, increasing throughput and expanding rule set sizes including the incorporation of novel network functions. In comparison to an FPGA-based approach providing 3.9 Gbps at 630 mW, the memristor-based TCAM provided 47.2 Gbps at 1.3 W. The study concluded that the memristor-based TCAM cells in combination with Memristor Random Access Memory (mRAM) circuits can be particularly beneficial for large scale networks due to superior energy and performance gains.

b: NETWORK INTRUSION DETECTION

In a follow-up research, Graves et al. [73] proposed a memristor-based TCAM architecture for network intrusion detection. The authors proposed a TCAM cell design for intrusion detection in RegEx applications. Improvements in power consumption, throughput and compression techniques were achieved by employment of nanoscale memristors in TCAM cells. The study showed that the use of memristors can enhance throughput by four times (upto 8 Gbps) with only 55% power consumption for network intrusion detection. Moreover, the use of *striding* can further increase the performance in terms of throughput, if required.

c: CMOS/MEMRISTOR HYBRID ARCHITECTURE

Hybrid use of CMOS-based transistors and memristors is the most feasible option for practical implementation of memristors. In this regard, Tabassum et al. [74] presented a memristor-based TCAM cell employing twelve transistors instead of sixteen, by using memristors for TCAM. The results suggest that the memristor-based TCAM consumes 70% less energy, 43% less search time and 27% less area. The study concluded that the proposed NOR type implementation can be a milestone for future electronic applications based upon the performance. However, the scope of this study is limited to the electronics of cell design and it needs further research for implementation of complex network processing functions.

3) LESSONS LEARNED

The review of previous researches shows that some studies proposed and designed the architecture of a TCAM cell with a focus on the hardware and electrical properties of memristors. Moreover, use of memristor as a storage element showed promising performance due to non-volatility and state-based response. The design of CAM was also realized using memristors and it also showed an increase in performance as compared to the state-of-the-art designs in terms of energy efficiency and implementation of complex network functions. Moreover, application-specific scenarios showed that the memristor-based TCAM can be used for regular expression matching and network intrusion detection systems. Also, based upon the CMOS architectures, performance and benefits of memristor-based TCAM can be increased with an integration to current components by combining CMOS and memristor-based architectures. The take away from this line of research includes the realization and feasibility of TCAM using memristors. This avenue of research showed that the memristor-based TCAM can be implemented with better performance than the traditional TCAM architecture.

Algorithm 1 Memristor-Based TCAM Search and Write Operation

```
Entries E = [E_1 ..... E_N]
Function tcam(Q, O)
    Data: Query and operation
    Result: Search location or write completion
    if O=='Search' then
        rows_search(Q)
        while rows_search(Q) do
            Out = match_query(Q)
        end
        return Out
    else if O=='Write' then
        search_write_location(Q)
        rearrange_entries(E)
        write_match_table(Q)
    end
end
```

D. NETWORKS PERSPECTIVE OF MEMRISTIVE TCAM

The computer science perspective of the hardware operations of TCAM is expressed in Algorithm 1 in form of the TCAM search operations. The *tcam()* function performs the search and write operations by using query Q and operation O, where Q refers to the pattern being searched and O refers to the read or write operation. During search operation, all rows of TCAM are searched for the query Q and the matched locations are returned in the variable *Out*. On the other hand, the write function of TCAM performs the write operation in decreasing order of priority list of stored rules inside the match table. Hence, *search_write_location()* function performs the location search and *rearrange_entries()* adjusts the rows of match table to create the location for new entries. Lastly, *write_match_table()* stores the entry inside the TCAM memory for the match table.

The network packet processors can use the memristive TCAM for a number of operations including IP lookup, intrusion detention and mapping large number of policies for the Internet packets. The packet processors are the most promising entities benefiting from the memristive TCAM because of high speed data processing requirements. Moreover, in-network operators can incorporate the memristive TCAM for adding cognitive functionality inside the packet processors. The benefits of TCAM can be extended to replace the traditional network memories with the TCAM because memristive TCAM performs all search operations within one clock cycle and consumes less energy than the traditional technologies.

V. LEARNING SYSTEMS

The current design of computer networks requires data operations in the path between source and destination with little energy consumption and complex cognitive functions. Although, memristors provide the state-based energy efficient function for modeling a complex response, but, coupling of memristors is a fundamental requirement for incorporation of complex network functions and cognitive functions inside the network. This coupling of memristors can be achieved in various configurations based upon the desired network functions such as reconfigurable architectures, reservoir computation architectures, neural network architectures and neuromorphic computing architectures. In reconfigurable architectures, programming of memristors is utilized to reprogram the state of the memristor and coupling arrangement is based upon the required functions. Crossbar arrays and FPGA designs are some of the prominent examples in this category. On the other hand, reservoir computation empowers the memristive connectivity with cognitive functions by mapping a range of output functions to a given set of input signals. The training of memristive reservoir architecture can be performed by reprogramming the individual conductances of memristors inside the reservoir. In neural networks, the programming of memristors is utilized similar to the weight function in a neural network model. While, neuromorphic computing utilizes the synaptic behavior and memristive cognitive properties similar to the human brain for modeling of cognitive functions. In this section, we review researches in the domains of these four architectures in order to perform cognitive decisions in the future packet processors built upon memristors.

A. MEMRISTOR-BASED RECONFIGURABLE ARCHITECTURES

In this subsection, we present the memristor-based novel architectures focusing on gate array architectures, reconfigurable architectures, FPGA designs and application-specific architectures including deep packet inspection and data security architectures in reference to their applications in networks.

1) GATE ARRAY ARCHITECTURE

Computational architectures can be designed using the gate architectures with programmable threshold control logic by leveraging from the state-based response of memristors. Rajendran et al. [75] proposed the gate array architecture by using memristors as weights at the inputs of the threshold gates and authors considered the power, area and delay considerations for memristor-based designs. In comparison to the traditional lookup table (LUT) architecture, the proposed architecture decreases power and area consumption by 75%. On the downside, the combination of a large number of memristors in the gate array architecture increased the delay of the system which is a challenge for future research.

2) FPGA AND CSGR ARCHITECTURES

The reconfigurability and programmability of hardware can assist in the computational performance by tuning the hardware according to the software requirements and this task can be achieved by reconfigurable architectures. FPGA and Coarse grained reconfigurable (CSGR) architectures are two major types of reconfigurable architectures with extensive applications in computational tasks like pattern matching and image recognition. Memristors, as two terminal resistive switches, can transform the reconfigurable architectures by providing programmable parameters for the required software functionality. Lu et al. [76] studied the behavior of memristors for reconfigurable architectures and showed better switching time, switching ratio, endurance and retention for memristor-based architectures. Extensive device modeling in SPICE concluded that the use of memristors can lead to higher function densities due to a range of programmable states and it also provides power efficiency due to less computational operations and memristor's limited power requirement. The scope of this research limited to small scale setup built upon simplified circuits and authors left the large scale analysis built upon complex circuits for the future research.

3) FPGA ARCHITECTURES

Memristors can revolutionize the FPGA architectures by providing control logic for the network algorithms. Cong and Xiao [77] proposed a novel memristor-based FPGA architecture, *mrFPGA*, using only memristors and metal wires for interconnection. In the proposed design, memristors are helpful in providing the capacitance shielding effect from unused routing paths and decreasing the delay of the system. The authors also proposed an adaptive buffer mechanism to further increase the performance of memristor-based architecture. The study concluded that mrFPGA provides area, speed and power improvements by 5.18, 2.28, and 1.63 times, respectively, in comparison to 20 largest MCNC circuit benchmarks.

4) CROSSBAR NEUROMORPHIC ARCHITECTURES

Memristor-based crossbar neuromorphic circuit is specially useful for achieving application-specific tasks like Deep Packet Inspection (DPI) on mobile devices requiring area, power and throughput efficiency. Bontupalli et al. [78] presented two types of crossbar array neuromorphic circuits for static pattern matching and regular expression circuits. The study showed that throughput up to 160 Gbps can be achieved through memristor-based crossbar array architectures with greater accuracy and lower latency. On the contrary, alternate DPI standards (including *de facto* Snort) provide less than half of this throughput.

5) SECURE CROSSBAR ARCHITECTURE

Memristor-based non-volatile storage architectures require special attention in handling security and privacy constraints especially in the area of personal computing and IoT. Kannan et al. [79] studied the security challenges and proposed a Sneak-Path Encryption (SPE) technique based architecture for memristor-based memory applications. Study suggested the use of data encryption for memristor-based crossbar array architectures by incorporating critical memristor parameters including physical characteristics, multi level cell capability and sneak paths in crossbar memory architectures. SPE proved to be resilient under the application of three challenging attack scenarios. However, the major limitation of the proposed technique is its overhead in form of 16 delay cycles and 1.5% degradation in performance.

6) LESSONS LEARNED

A review of memristor-based novel architectures showed that memristors can be utilized for the development of programmable threshold gates which can aid in the process of logic development. The programmability of memristors can be used to develop new architectures employing a range of states for various network functions. The current network paradigms like P4 [18], NetFPGA [80], etc., also build upon same basic principles of reconfigurability and memristors can aid in the development of complex architectures built upon small reconfigurable memristive circuits. This results in not only direct performance enhancement due to efficient operations, but, also atomic operations of memristors provide energy efficient response as compared to the transistors. Memristors showed performance gains for reconfigurable architectures due to their prominent switching, timing and energy consumption statistics. Packet processors can benefit from the reconfigurable architectures in form of hardware accelerators and logic synthesizers by employing fast switching and less energy consumption properties of memristors. It also results in a direct performance improvement in terms of throughput in the network due to less complex operational overhead in the packet processors. In computational applications, DPI has also been facilitated with great performance (in terms of throughput) by employing memristor-based novel crossbar array architectures and it can be used on network routers for combating network attacks by identifying attackers' signatures. Lastly, owing to the state-based non-volatile characteristics, memristors can replace transistors in the FPGAs and ASICs, and they can be reprogrammed through a range of states for complex logic building with less energy consumption.

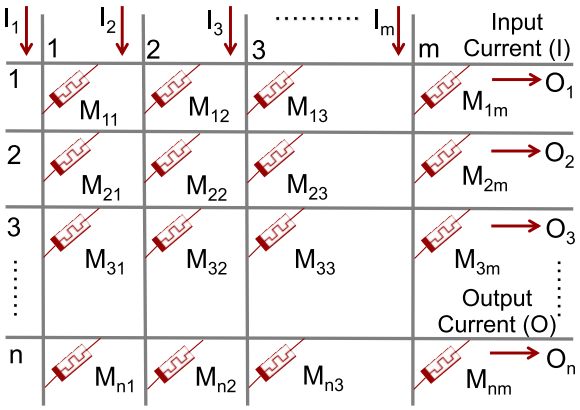

FIGURE 10. Memristor-based crossbar array architecture.

7) EMPLOYMENT IN PACKET PROCESSORS

The findings of previous researches can be extended to the packet processors by reviewing the memristor-based crossbar array architectures as shown in Fig. 10. For an $m \times n$ crossbar array connecting m input lines to n output lines, $m \times n$ memristors are required where each memristor connects one input line to one output line. The output current O is an array of currents obtained from the corresponding n output lines. By using memristor programmability, we can vary the conductance inside the memristor. The response of the given network architecture can be demonstrated by using the current voltage relationships. Using Ohm's law, voltage (V) and current (I) are related to the resistance (R) of the memristor by (1).

$$V = IR \tag{1}$$

We know that the conductance (G) is inversely related to the resistance of the memristor as shown in (2),

$$G = 1/R \tag{2}$$

According to the Kirchhoff's current law, sum of input currents is equal to the sum of the output currents through any node. Hence, the output current from the first output line is given by (3). Equation (4) shows the array of currents from all n output lines.

$$O_1 = G(M_{11}) \times I_1 + G(M_{12}) \times I_2 + G(M_{13}) \times I_3 + \ldots + G(M_{1n}) \times I_m \tag{3}$$

$$O = [O_1\ O_2\ O_3\ \ldots\ldots\ldots\ldots\ O_n] \tag{4}$$

By programming the conductance G of the memristor, we can program the memristor-based crossbar array architecture for a range of functions ranging from basic algorithmic computational operations to complex functions like in-network complex event processing and cognitive decision making.

B. MEMRISTOR-BASED RESERVOIR COMPUTATION

Reservoir computation is a computational framework for mapping input signals to higher dimensional computational spaces by using a nonlinear fixed system called the reservoir. The background of reservoir computation stems from recurrent neural networks and it can be used to map a variety of complex network functions in the context of programmable nanoscale components, memristors. In this subsection, we review the memristor-based reservoir computation architectures with a focus on application-specific designs including big data analysis, feature extraction and classification, and image recognition.

1) PATTERN CLASSIFICATION ARCHITECTURE

Reservoir computation based upon memristors is a promising avenue of research due to the memristor's abilities for large scale packing, nonlinearity and non-volatile memory characteristics. Kulkarni and Teuscher [81] presented the first memristor-based reservoir computation architecture for computational tasks and demonstrated the feasibility of the proposed design for pattern classification and associative memory tasks. The authors used memristive programming features to test the performance of reservoir computation architectures. Simulations and evaluations were performed in Ngspice and *Genetic Algorithm* was used for training of the software framework. The study suggested similar performance gains for future work on complex cognitive functions and incorporation of memcapacitors and meminductors can further increase the performance margins.

2) ANOMALY DETECTION ARCHITECTURE

In another study on memristor-based reservoir computation, Sayyaparaju et al. [82], [83] presented a memristor-based crossbar array architecture that has been trained using STDP features on a *Wisconsin Breast Cancer* data set. A major improvement is the reconfigurability of the proposed architecture for handling anomalistic behavior. In this research, authors presented a robust liquid state machine implementation technique using memristor-based spiking neural network. The study showed that device level switching asymmetry issues can be successfully resolved using the proposed memristor-based architecture. However, the proposed architecture needs to be evaluated and validated for packet processors considering the network flow guarantee requirements which are challenging to achieve in a Spiking Neural Network (SNN)-based network.

3) REAL-TIME DATA ANALYSIS ARCHITECTURE

A fundamental requirement of neuromorphic computing is an in-depth understanding of the underlying mechanisms inside the nervous system. However, large amount of neuronal activity cannot be processed in real-time. Zhu et al. [84] proposed a memristor-based reservoir computation architecture for real-time analysis of large data sets including neuronal behaviors. Perovskite-based memristor was used for emulation of neuronal spikes and state of memristor captured the temporal features in the neuronal spike train. The study showed that memristor-based reservoir computation can emulate the processing with significant resemblance and sophisticated cognitive functions can be modeled using memristors. The scope of this study for real-time data analysis can be very useful for network applications with high real-time traffic demanding sophisticated data analysis at the compute nodes.

4) FEATURE EXTRACTION ARCHITECTURE

Feature extraction and analysis is one of the critical tasks in reservoir computation for big data analysis. Cai and Lu [85] presented the feature extraction and analysis methodology for memristor-based networks. The authors presented a hardware implementation of sparse coded algorithm on a 32×32 memristor crossbar architecture and emulated the characteristics of a neural network with better optimal settlement and showed the reconstruction of features in a memristive array. The results on a range of data sets including MNIST and real world images show that the proposed architecture can identify the hidden features. Moreover, authors showed successful reconstruction of second order nonlinear dynamical systems using the proposed memristive reservoir computation architecture.

5) IMAGE RECOGNITION ARCHITECTURE

Image recognition on the edge computing nodes is a critical network challenge for the current Internet. Ran et al. [86] developed a memristor-based blaze block circuit for edge computing system with a specific focus on image recognition. The design includes a memristive convolutional neural network followed by multiple modules of memristive pooling and block elements for the operation. The simulation results showed an accuracy of 84.38% for the CIFAR-10 data set with performance benefits in energy consumption, and time

and area efficiency. The research also studied and showed the performance improvement by incorporation of multi-state memristor conductance and data quantization effects. A shortcoming of the research is the lack of optimization in memristive and computational resources deployed in the design process.

6) LESSONS LEARNED

The review of previous studies suggested that the memristor-based reservoir computation architectures are feasible and complex learning functions can be developed through such architectures. The capabilities of memristors for nano-packing, state-based response, non-volatility and less energy consumption made them an ideal component for reservoir computation architectures. The development of novel learning functions can aid in deploying cognitive functions at packet processors and end nodes without any data movement requirement between memory and computational units. In this regard, several researches showed the performance benefits for memristor-based reservoir computation architectures in mapping large number of functions including data analysis for network components. Also, novel complex crossbar array architectures can be deployed for network functions for edge computing and data center computing. Prominent applications of reservoir computation include data analysis which motivated researchers to look into feature extraction and analysis in a memristive architecture. Another prominent application includes image processing and application of blaze block circuits with an implementation perspective of memristors. All of these applications suggest the use of memristors for energy efficient state-based cognitive network functions inside packet processors.

C. MEMRISTIVE NEURAL NETWORKS

The requirement of cognitive network functions has motivated the use of neural networks inside the packet processors. A cognitive network function can be deployed in network middleware entities or edge devices to handle the traffic flows based upon the cognitive requirements of network functions like load balancing, flow management, routing, firewall development, etc. However, the application of current neural network approaches is challenging due to the data movement between memory and computational units, huge energy consumption and limited state options in the transistor-based implementations. In this regard, the state-based response of memristors with programmability features provides an alternate to the traditional transistor-based neural network architectures. At the same time, less energy consumption of memristors can help in achieving complex AI operations at end nodes and edge devices. An energy efficient neural network architecture can ultimately replace the state-of-the-art network paradigms like P4 [18], NetFPGA [80], etc., due to the cognitive handling of packets unlike any other architecture. In this section, we would highlight the previous researches and their findings in the domain of memristor-based neural networks. Fig. 11 presents an overview of the research dimensions and key research areas in the studies on memristor-based neural networks.

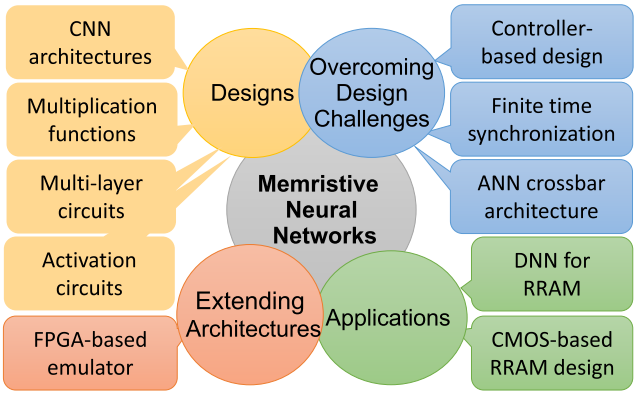

FIGURE 11. Research in memristive neural networks.

1) MEMRISTOR-BASED CNN

Memristors can be used for establishing neighboring connections in Cellular Neural Networks (CNNs). Lehtonen and Laiho [87] proposed a novel memristor model for CNNs and evaluated the results in the SPICE simulation environment. The authors concluded that memristors can provide all the desirable functionalities in the CNNs including linear I-V curves in the compute mode and steep programming thresholds. It was shown that the template variation is less feasible in memristor-based CNNs, and cascades of memristive networks were proposed for changing the templates.

2) FINITE TIME SYNCHRONIZATION FOR RECURRENT NEURAL NETWORK (RNN)

Memristor-based networks pose a significant challenge to timing synchronization in neural networks due to the variable processing delay among various paths in the network. Gao et al. [88] studied the memristor-based recurrent neural networks in reference to finite time synchronization. The authors proposed a novel controller-based network which can decrease the synchronization time by incorporating the differential inclusions theory and set-valued maps. Analytical and simulation results demonstrated the effectiveness of the proposed controller for achieving finite time synchronization. This study aided in the practical implementation of memristor-based neural networks.

3) MULTI-LAYER CIRCUIT DESIGN

Despite various promising results, practical implementations require circuit design for memristor-based implementations. Zhang et al. [89] proposed a memristor-based multi-layer circuit design using a single array of memristor for all kinds of synaptic responses. The authors' contributions also span in the use of memristors as switching elements and development of adaptive back propagation algorithms which can be used for Exclusive OR (XOR) operation and character

recognition. The research concluded that the proposed design provides higher recognition rates in fewer clock cycles by use of memristor-based design approaches. Also, increasing the noise from 0% to 30% decreases memristor-based network's performance from 100% to 97.4%, while, the performance in original design (without memristors) decreases from 99.8% to 72.5%. Hence, the memristor-based design also proved to be resilient to system noise which suggests the use of memristors in complex circuits requiring large scale replication of small scale circuits.

4) MEMRISTIVE ACTIVATION CIRCUIT FOR DEEP NEURAL NETWORK (DNN)

Implementation of Deep Neural Networks (DNNs) requires a memristive activation circuit. In this regard, Bala et al. [90] presented a *MIN* function-based circuit for memristor-based neuromorphic implementation. Two memristors and a comparator were used for realization of a Rectified Linear Unit (ReLU) activation function. The given circuit was simulated and performance was compared to the traditional crossbar array architectures. The results showed that the proposed architecture gives better performance than the previous state-of-the-art architectures in terms of time and area efficiency. Moreover, the training process also takes less time by using the proposed architecture because of the simplicity of the memristive activation function and training inside the network without any data movement between memory and computational units.

5) STABLE NEURAL NETWORK WITH CONTROLLER

Instead of the traditional decentralized approaches, a novel class of neural networks can also be built upon the memristors. Wu and Zeng [91] focused on achieving exponential stabilization and formulated the conditions for using memristors in neural networks. Moreover, the authors proposed a novel controller along with the estimation of theoretical results for development of a neural network and showed significant performance gains due to the presence of a centralized controller. The decrease in cost function in memristor-based neural networks was the major contribution of the authors.

6) DESIGN OF MULTIPLICATION FUNCTION

The design, analysis and application-specific focus of memristor-based CNNs can chalk out the feasibility of cognitive devices for networks. Duan et al. [92] suggested that the memristor-based multiplication function can be much more efficient than the traditional transistor-based design. Moreover, the nonlinear current-voltage characteristics, negative differential resistance, non-volatility, high density, and programmability of synaptic weights were the beneficial features of the proposed CNN design. Monte Carlo simulations were used to analyze the proposed design and image processing functions greatly benefited from the proposed approach.

7) MEMRISTOR NETWORK EMULATOR USING FPGA

The use of memristors can also be employed in conjunction with an FPGA platform. Vourkas et al. [93] made the first effort to implement a digital memristor emulator on an FPGA platform. The authors used the threshold type bipolar memristor model and proved the synaptic properties of the memristor. However, a shortcoming of the research includes its limited scope on the issues of scalability and complex memristive models for ANN.

8) DNN FOR RESISTIVE READ ACCESS MEMORY (RRAM) ARCHITECTURE

DNNs have extensive applications in Resistive Read Access Memory (RRAM) design by employing in-memory computing architectures. Wang et al. [94], [95] discussed the possible implementations and design proposals for implementation of DNN in RRAM architecture. Critical factors including finite array size, quantized partial products and non-ideality factors were evaluated and techniques were proposed to enhance the performance of DNNs in RRAM. The study showed that RRAM architecture based on DNN can successfully deploy the models like VGG-16, MobileNet and Recurrent Neural Network (RNN)/LSTM on ImageNet dataset and perform the text classification tasks.

9) CMOS-RRAM ARCHITECTURE

In a follow up research, Correll et al. [96] studied the semantics and architectures for the analog design of a CMOS-RRAM reprogrammable neuromorphic chip. By employing RRAM, the bottleneck for data availability in data hungry applications, including AI and machine learning applications, can be bypassed. The authors proposed the first prototype 54 $\times$ 108 RRAM crossbar array hierarchy followed by its digital and analog circuitry. They used Analog-to-Digital Converters (ADCs) and Digital-to-Analog Converters (DACs) to present a hybrid CMOS-RRAM architecture which demonstrated performance improvement for machine learning functions including online learning. The study concluded that the proposed RRAM provides gain in throughput (2.6 Giga Operations Per Second (GOPS)) and reduction in power (307 mW for combined CMOS-RRAM).

10) MEMRISTOR's NON-IDEAL BEHAVIOR

Hybrid use of memristors and CMOS is a feasible implementation option due to the CMOS resilience and track record of feasible components built upon CMOS. Pham and Min [97] studied the critical aspects relating to the non-ideal behavior of memristors-CMOS hybrid usage. The non-ideal behavior is much related to the crossbar connection of memristors resulting in variations in synaptic memristance, parasitic resistance and physical conditions. The non-ideal characteristics were demonstrated on ReLU and Sigmoidal activation functions. This study laid the foundation for future usage of memristors in hybrid networks and identified these shortcomings in the hybrid use of memristors and CMOS.

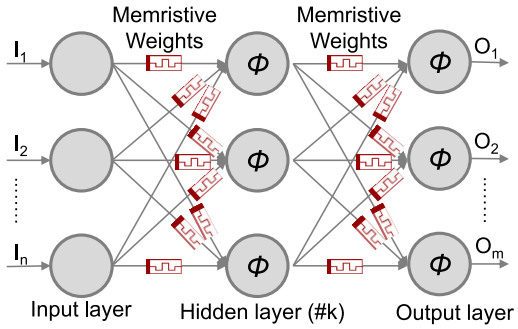

FIGURE 12. Memristive neural network for network packet processors.

11) LESSONS LEARNED

The review of memristor-based research in neural networks shows that memristors can be deployed for neural network architectures. Research over network design showed that the memristor-based ANN and CNN circuit designs required less training time and energy due to the colocalization of memory and computation. Also, application of complex neural network functions including multiplication function and multi-layer computation showed promising gains with memristors due to memristors inherent state-based response. Moreover, novel architectural designs including FPGA-based emulators and activation circuits for DNN were also realizable with memristors. Stability has been a challenge in memristor-based neural networks, but, this shortcoming can be bypassed by moving towards CMOS-based hybrid architectures or employing a centralized controller-based neural network. On the application side, memristors along-with CMOS presented suitable alternatives for DNN with some critical challenges, but, the benefits of memristive architectures promote their usage for complex neuromorphic architectures.

12) MEMRISTIVE NEURAL NETWORKS IN NETWORK PACKET PROCESSORS

In reference to the network-based applications used in packet processors, memristor-based neural networks can be modeled as shown in Fig. 12 and Algorithm 2. In a memristive neural network, every node receives input current (I) from the connected input side through memristors. Using Ohm's law, the currents at the outputs of memristors are a function of the resistance (inverse of conductance (G)) programmed inside the memristor. Using Kirchoff's current law, the current at the output of the node is the sum of the input currents. Moreover, neural network can add a bias (b) and an activation function (ϕ) at the compute node to make more intelligent and controlled decisions. The activation function and bias can change the behavior of the circuit and can be tuned based upon the back propagation algorithm and required constraints. In networks, this configuration can be used to increase the range of decisions for in-network processing at packet processors. During in-network processing, packet processors can base their decisions upon a range of network functions incorporating load, congestion, routing and administrator policies, which can ultimately result in cognitive handling of the network packets.

Algorithm 2 Memristor-Based Neural Network

```
Current I = [I_1 ..... I_N]
Conductance G = [G_1 ..... G_N]
b;//Neural network bias
φ;//Activation function
Function neural_network(I, G)
    Data: Currents and conductances at nodes
    Result: Weighted average followed by biases and
            activation functions at nodes
    x = I × G^T;
    output = φ(x + b);
    return output;
end
```

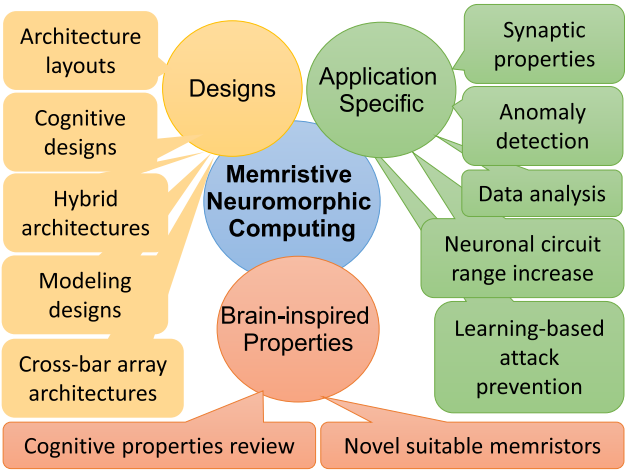

FIGURE 13. Research in memristive neuromorphic computing.

D. MEMRISTIVE NEUROMORPHIC COMPUTING

Neuromorphic computing emulates the neuron spiking capability of human brain along with the colocalization of memory and computation for achieving superior benchmarks in energy and cognitive capability [98]. In this section, we review the neuromorphic computing architectures based upon the memristors and present a glimpse of the ongoing research as depicted in Fig. 13.

The limitations of transistor-based components, in terms of energy efficiency and computational models, motivate the research community to study the use of novel components including memristors for neuromorphic computing [99]. In this regard, the construction of electronic neurons resembling the synaptic capabilities of human brain is the most crucial step and a number of studies focused over the use of memristors for neuromorphic architectures [100]. Upadhyay et al. [101] surveyed the synaptic properties and implementations of these properties in neuromorphic

architectures. The authors reviewed the decision mechanisms of human brain and compared them to the properties of new memristive devices for emulating neuronal behavior. The study showed that capturing neuronal activity in an integrated memristor circuit is a major challenge for neuromorphic architectures due to the brain's complex communication mechanism. In another research, Zhang et al. [102] reviewed the synaptic properties of memristors for use in neuromorphic computing. The review suggested that area and energy efficiency followed by applications of neuromorphic computing are key areas for memristor applications. The authors motivated the feasibility and applicability of memristor crossbars for DNNs, ANNs and SNNs. Challenges in the design process include the selection of suitable memristors considering their hardware properties and ideal crossbar connections for resembling the properties of brain. In the below lines, we would review the research on memristor-based neuromorphic architectures.

1) NEUROMORPHIC ARCHITECTURE DESIGN

Indiveri and Liu [103] presented a brain inspired neuromorphic computing architecture by using transistor-based CMOS technology. Memory and computation were co-localized to emulate human brain. The design incorporates the synchronous and asynchronous modes of operation to present performance benefits of neuromorphic computing. However, authors left the use of nanoscale devices (including memristors) for future research, but, they presented a baseline by comparing and analyzing neuromorphic computing architectures with the traditional architectures.

2) MEMRISTOR-BASED NEUROMORPHIC ARCHITECTURE

Owing to the drawbacks of CMOS-based transistors, many researches have focused over the use of nanoscale components for future cognitive architectures. Liu et al. [104] presented a novel framework for use of memristors in neuromorphic computing. The study showed that the spiking designs with digitized interfaces are more energy and cost efficient than the counterpart analog designs. Owing to the analog design benefits in performance and cost, the authors proposed a current sensing scheme to improve the analog design. Also, novel hardware-based strategies were suggested to remove the inherent memristor defects and upto 99.3% performance gains were obtained by using the proposed strategies.

3) MEMRISTOR-BASED COGNITIVE ARCHITECTURE

The development of a cognitive architecture is a critical task for incorporating learning capabilities in the packet processors. Zheng et al. [105] proposed a novel memristor-based computing architecture leveraging from the high learning and memory capabilities of memristors. The authors developed a neural network architecture by incorporating spike timing and rate dependent plasticity, and generating spike behavior similar to the functioning of human cortex. The study showed through simulations that the neuromorphic architecture can provide better performance in terms of cognitive decisions than the conventional architectures.

4) MEMRISTOR-BASED CROSSBAR ARRAY ARCHITECTURE

It is crucial to model the crossbar array design using memristors for employment in neuromorphic architectures. Hu et al. [106] proposed the use of memristor crossbar arrays for hardware implementation of brain state-in-a-box neural networks. The authors developed a mathematical model for training and recall features. Impact of various noise levels through extensive Monte Carlo simulations proved the resiliency of the proposed neuromorphic model. Multi answer character recognition algorithm was used for robustness analysis of the proposed design. The research concluded that the correlation between memristor crossbar arrays and optimization of summing amplifier are the most critical factors in the neuromorphic circuit's performance.

5) MEMRISTOR-BASED HYBRID NEUROMORPHIC ARCHITECTURE

An interesting practical line of research is the incorporation of memristor-based architectures in the traditional networks using hybrid design topology. In this regard, Liu et al. [107] proposed a computing architecture, called *Harmonica*, by incorporation of memristor-based neuromorphic architecture in the traditional architecture. The authors suggested the use of mixed signal interconnection network to increase the computational power of ANNs. The research showed that Harmonica provided performance and energy gains by 27.06× and 25.23×, respectively. Moreover, incorporation of Multilayer Perceptron (MLP) can further increase the performance of the network. The performance gains of Harmonica proved to be much better than the traditional designs with digital neural processing units or memristor-based crossbar arrays. Major driving factors for high performance statistics included the high throughput in mixed-signal computation, reconfigurability of the network, low data overhead and concise coordination interface.

6) RANGE INCREASE IN NEURONAL CIRCUIT

On the side of hardware implementation, a critical problem is the increase in range of the neuronal circuit. Jiang et al. [108] proposed a cyclical sensing scheme for increasing the range of high speed Integrate-and-Fire Circuit (IFC). Simulations showed that cyclical sensing IFC performed matrix multiplication with an integration of 32 × 32 memristor crossbar arrays. Power efficiency was also achieved for large input current and high output frequency.

7) MEMRISTOR CROSSBAR ARRAY

Practical deployment of memristor-based crossbar arrays is a major line of research for feasibility analysis. Kataeva et al. [109] designed a memristor-based crossbar chip with 2.4 million memristors. The authors used 1032 48 × 48 crossbars of Al_2O_3/TiO_{2-x} memristors and performed the analysis by integrating the chip with an FPGA board.

Testing of the chip on an FPGA board showed that it can be successfully employed for the deployment of ANNs using memristors.

8) VON NEUMANN VS MEMRISTOR-BASED NEUROMORPHIC ARCHITECTURE

Comparison of the memristor-based architectures with traditional design techniques can signify the gains in the performance improvements. An et al. [110] compared the conventional von Neumann architecture with the neuromorphic architecture. The authors proposed a memristor-based neuromorphic architecture and estimated its energy and performance efficiency. Three variations of neuromorphic architectures were proposed which were emulating the functioning and spiking capabilities of human brain. Authors varied the neuronal locations, connectivities and distribution patterns in order to analyze the performance of memristor-based neuromorphic architectures in various designs. The study showed that the neuronal architecture can mitigate the congestion issues in the network, and amplify the performance of data analytics during in-network processing.

9) MEMRISTOR MODELING FOR NEUROMORPHIC DESIGNS

Inspired by the second order memristors due to synaptic plasticity characteristics, Marrone et al. [111] developed a model to link memconductance and temperature to synaptic efficacy and temperature. The authors also developed a model to link the second order memristors to the neuromorphic architectures and compared them with the spike and plasticity models. The proposed model was analyzed for varying factors including spike pairs, triplets and quadruplets at different frequencies and the model's behavior was similar to the brains neural network. The model laid the foundation for future research in the employment of second order memristors for neuromorphic computing.

10) LEARNING-BASED ATTACK PREVENTION

Neuromorphic learning systems based upon memristors are susceptible to learning-based attacks due to the memristors state-based response and various studies focused over this issue. Yang et al. [112] presented a model to prevent an attacker from learning the behavior of the memristor-based neuromorphic architecture. The proposed strategy employs the memristor obsolescence effect for users where only legitimate users get complete access to the network. The resilience of the proposed scheme was demonstrated on four data sets with variations in constraints, cost functions and activation functions. The proposed scheme provided an increase of nonlinearity index by 179.93% and 288.99% on Mean Square Error (MSE) and error rate, respectively.

11) ANOMALY DETECTION USING NEUROMORPHIC COMPUTING

The use of memristors in neuromorphic computing can improve the detection and estimation accuracy in complex large scale data analysis. Chen et al. [113] proposed the use of memristors for neuromorphic computing by focusing on anomaly detection in large traffic networks. The authors used the cogent confabulation model in inspiration from the human neocortex system. Likelihood ratio test was used to identify the abnormal vehicles demonstrating an anomalistic behavior. The research showed that the vehicles over speeding, tailgating and stopping-and-going can be identified with an anomaly score ranging from 25% to 100% depending upon the particular conditions and circumstances.

12) MEMRISTOR-BASED ANN FOR DATA ANALYSIS

Memristor-based ANNs can have applications in user security and privacy. Fu et al. [114] performed research on noise injection in the memristor-based ANNs. The authors proposed a linear optimization strategy which updates the memristor conductances analogous to the weights in memristor-based ANNs. Test results on *MNIST* handwriting data set demonstrated an increase of accuracy by 39.67%. The study concluded that the memristor-based ANN can provide higher user privacy for large scale data analysis than the conventional designs.

13) NOVEL MEMRISTOR TYPES FOR NEUROMORPHIC COMPUTING

The study of hardware characteristics of novel memristor types is a prominent research direction in memristor-based neuromorphic architectures. In this regard, Zaman et al. [115] conducted an experimental study to chalk out the suitability of memristive devices for neuromorphic applications. The authors studied the *GeTe* and VO_2 memristors, and analyzed the characteristics of these memristors including symmetry, reliability, stability and programmability. The study showed that the programmable resistance range can be utilized for the development of neuromorphic architectures. The research showed that high variability in device characteristics can be efficiently utilized to map a variety of neuromorphic functions.

14) NOVEL MEMRISTORS FOR NEUROMORPHIC CHIPS

Memristors exhibit significant characteristics quite useful for employment in neuromorphic architectures. Wang et al. [116] showed the resistance programming of a new type of memristor, namely $LiNbO_3$ memristor, with significant characteristics, including tunability and switching curves, which motivate its usage in neuromorphic systems.

15) REVIEW OF MEMRISTOR-BASED CIRCUITS

The development of a memristor neural circuit is a major challenge for neuromorphic architectures. In this context, Zhao et al. [117] reviewed the memristor-based neuromorphic implementations for ANNs. The authors presented the challenges and suggested that the development of a basic neural circuit element followed by an extensive connectivity would be a major hardware challenge for neuromorphic architectures. However, memristor-based crossbar architectures

can promote the development of complex neuromorphic functions.

16) LESSONS LEARNED

The review shows that neuromorphic computing is the most complex and critical implementation for memristor-based architectures. However, implementation of memristor-based neuromorphic architecture is feasible and designs have been proposed by a number of researches. The scope of this line of research includes the design of memristor-based crossbar arrays using memristive synaptic properties for design and implementation of cognitive functions and architectures. Another prominent line of research includes the application-specific designs considering anomaly detection and data analysis using neuromorphic architectures. Memristor-based architectures can aid in the deployment of pattern recognition techniques. The previous studies have modeled the memristor-based neuromorphic architectures and mitigated the basic shortcomings of memristive neuromorphic architectures including prevention of feature learning attacks, designs of better performing crossbar arrays, research on suitable memristor types for neuromorphic architectures, increase in neuronal range and use of hybrid CMOS-memristor components. The review of neuromorphic architectures with memristors laid focus on the realizability of memristor-based architectures and the attributes and properties of memristors can be utilized to develop neuromorphic architectures.

17) NETWORKS PERSPECTIVE OF NEUROMORPHIC ARCHITECTURES

Neuromorphic computation can incorporate the cognitive decision mechanism in the current Internet components. The computer science analogous function for the neuromorphic architecture is shown in Algorithm 3. The function *neuromorphic_comp*(S) takes the binary spikes S as inputs with voltage V at high or low levels. At the first stage, function *synapse_ftn*() reshapes the synapses based upon the priority and required function characteristics. Later, the *sum*() function accumulates the spikes for a continuous period of time. If the summation of spikes is greater than the threshold voltage, then, an output spike is generated by the function *generate_output_spike*(). Also, potential resets after the generation of output spike. If the elapsed time is greater than the threshold time, then, the outdated spikes are discarded from the potential by *discard_outdated_spike*() function. Lastly, the function *neuromorphic_comp*(*spike_wait*(S_{new})) keeps repeating as long as an input is available at the receiver's end.

In networks, the asynchronous clock and colocalisation of memory and computation can transform the packet processors. All switches and network routing entities can base their decisions from a range of network functions including routing, congestion control, load balancing and administrative policies. The decision can be derived in less time due to the asynchronous clock which would eliminate the delay in the packet processor. Moreover, a major benefit of the neuromorphic computing is the increased resilience, security and privacy of the proposed system because all network packets can be addressed precisely based upon the situation. It can also decrease the load on the network by eliminating the need of control packets used by various protocols (messenger applications etc.) because neuromorphic architecture can itself cater for control packets by making intelligent and timely decisions.

Algorithm 3 Memristor-Based Neuromorphic Network

```
Spike_voltage V = [1, 0]
Spikes S = [S_1 ..... S_N]
Function neuromorphic_comp(S)
    Data: Spikes of neural activity
    Result: Output spikes after decision process
    O = synapse_ftn(S)
    potential = sum(potential,O)
    if potential ≥ threshold then
        generate_output_spike();
        reset_potential();
    else if time ≥ time_threshold then
        discard_outdated_spikes();
    end
    neuromorphic_comp(spike_wait(S_new));
end
```

VI. DISCUSSION

In this section, we present the challenges and directions of future research for memristor-based cognitive packet processors.

A. CHALLENGES IN MEMRISTIVE NETWORKS

Memristor-based networks can implement complex network functions in form of colocalization of memory and computations with applications in the domains of rule-based systems and learning systems using TCAM, reconfigurable architectures, reservoir computation architectures, neural network architectures and neuromorphic computing architectures. However, these benefits have an associated cost in terms of the software and hardware challenges, and complexities in the design process. In this section, we review the challenges for memristor-based networks along with the possible remedies for these problems.

1) SNEAK PATH CURRENTS

A major challenge for memristor-based crossbar array architectures is the unwanted current paths in the networks. In architecture designs, these current paths act as a system noise and are called Sneak path currents. Majority approaches to avoid sneak path currents focus on multi-stage

reading [118], development of an unfolded architecture [119], application of diode gating [120], employment of transistor gating [121], use of complementary memristors [122], application of nonlinearity characteristics of memristors [123], and use of Analog Current (AC) sensing [124]. Contrary to the previous approaches, Zidan et al. [125] analyzed the sneak path currents in a memristor crossbar array, evaluated the power consumption and compared the energy statistics with the preceding researches. The authors proposed a solution for avoiding sneak path currents by gating the memory cell with a three terminal memristor. The proposed approach showed better noise margin and array size scalability for sneak path currents. However, sneak path currents still pose a major challenge for novel architectural configurations.

2) HARDWARE CHALLENGES FOR MEMRISTOR CROSSBAR ARRAYS

The implementation of a large scale memristor crossbar array also poses challenges for neuromorphic chips and neural network accelerators in terms of the computational model implementations. Shin et al. [126] designed a large scale memristor-based crossbar array architecture and studied its characteristics including matrix vector multiplication aspects for neural networks. Extensive experiments showed that IR drop, crosstalk and ripple effects are the major hardware challenges in the implementation of nanoscale devices and these effects can reduce the signal strength and increase noise in the system. Hence, scaling aspects are quite critical for diverse applications of memristor-based architectures.

3) AGING EFFECT IN MEMRISTORS

A major challenge in memristor-based crossbar array architectures is the aging effect in memristors. During programming of conductances in memristors, the high voltage pulses damage the filament in memristors and it decreases the number of available conductance cycles in the long term operation. To counter this issue, Zhang et al. [127] proposed the use of software and hardware training, along with the past aging status, for estimation of neural weights in programming the neural network. Also, the authors suggested that the selection of lower weight (conductance) values can ultimately lead to an increase in the life of memristor crossbar arrays by up to ten times without any reduction in classification accuracy of neural network.

4) LIMITATIONS IN MEMRISTORS

Stable switching and performance are a major challenge for memristor-based upon novel materials. For efficient and optimal implementation, the study on novel materials is required which can provide promising characteristics for memristors in neuromorphic computing. In this regard, Wang et al. [128] reported a new memristor with composition of $Pd/Al_2O_3/TO_X/Ta$ and exhibiting bipolar analog switching behaviors which can efficiently exhibit the characteristics of an artificial synapse. The researchers showed all properties of neurons including STDP, paired-pulse facilitation and long term potentiation/depreciation on the device and it consumed only 50 pJ of switching energy per spike. However, more widespread analysis of novel materials is required for stable memristor characteristics.

5) LIMITATIONS IN MEMRISTIVE NEUROMORPHIC FEATURES

Requirements of spiking neural networks in neuromorphic computing poses significant challenges on the usage of memristors inside the brain inspired networks. Some predecessor researches worked over the development of novel stable and application-specific memristors spanning over various structural dimensions. Murray [129] developed an Ag-in-oxide memristor which gives properties similar to Ca^{+2} ions in brain and efficiently exhibits short and long term plasticity properties. In another research, Wang et al. [130] developed a novel $Ag/Ag : Ta_2O_5/Pt$ self-doping memristor and showed through experiments that the proposed memristor can efficiently emulate the brain spiking properties. Contrary to the researches promoting oxide-based memristors, Majumdar et al. [131] showed the cycle-to-cycle variability of oxide-based memristors and proposed a new type of memristor-based upon solution-processable ferroelectric tunnel junctions with $P(VDF - TrFE)$ copolymer barriers. The proposed memristor exhibited promising switching, energy and behavioral characteristics for neuromorphic architectures. Considering the information storage and synaptic characteristics, Dang et al. [132] proposed and fabricated a bio-degradable biomimetic synaptic memristor-based upon $W/MgO/ZnO/Mo$ components on a silk protein substrate. The proposed device can be dissolved in special solutions in seven minutes which provides an added security layer to the neuromorphic devices and can emulate the apoptotic process of biological neurons. However, more research is required into suitable materials for embedding brain liking synaptic response with information security and privacy in the memristors.

B. FUTURE DIRECTIONS FOR MEMRISTIVE NETWORKS

Memristor-based packet processors can provide better energy efficiency along with the colocalization of memory and computation for better performance. The past trends for the memristor-based research in computation and memory architectures are presented in Fig. 14. Previous trends show that memristive neuromorphic computing has been the most studied topic followed by the novel computational crossbar architectures and neural networks in the field of memristive computational research. Based upon the past trends, future directions of research are discussed in this section.

1) NOVEL NETWORK FUNCTIONS

The development of novel network functions built upon novel components and materials is a prominent line of research. It is pertinent to mention that the packet processing architectures depend highly upon the conventional components and network functions which are energy hungry with lower

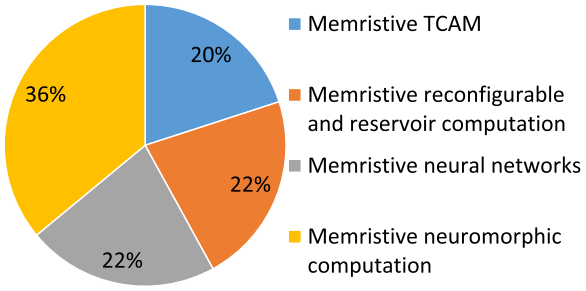

FIGURE 14. Classification of memristor-based networks research.

performance gains. By introducing new components, state-of-the-art protocols face challenges in coping with the new functionalities. As a result, new protocols and novel network functions built upon new architectures are required to harness the benefit of cognitive components.

2) FUTURE COGNITIVE PACKET PROCESSORS

Research on future cognitive packet processors, by incorporating AI and benefiting from novel self learning network functions, is another line of research. Neuromorphic computing and neural networks are already trying to emulate the human brain by developing complex learning functions and research on novel architectures can strengthen this line of research. Moreover, shift from synchronous to asynchronous clock inside a computer architecture and harnessing the colocalization of memory and computation inside a single unit is another major challenge. Hence, research on novel packet processors is required which can deviate from von Neumann architectural configurations and develop energy and performance efficient architectures.

3) NOVEL FUNDAMENTAL COMPONENTS

Similar to the resistors with memory (memristors), significant performance attributes are expected for capacitors and inductors with memories (namely memcapacitors and meminductors) [133]. In this regard, Jeltsema and Schaft [134] investigated the practicality of ideal memcapacitors and meminductors. The authors proved through mathematical evaluations that ideal memcapacitors and meminductors violate the first law of thermodynamics, which make their existence infeasible. The study suggests the use of non-ideal *memcapacitors* and *meminductors* which are not a source of free energy harvesting, but, their existence is much more practical and their properties can be utilized to make high performance devices. Perhaps, more research into novel fundamental components is a promising future research direction and it can chalk out the feasibility of energy efficient and cognitive packet processors.

4) NEW RESISTIVE ARCHITECTURES

The research on novel resistive architectures is a critical requirement to harness the full potential of resistive components. Lee et al. [135] conducted a thorough review and analysis of shortcomings of the previous non-resistive architectures

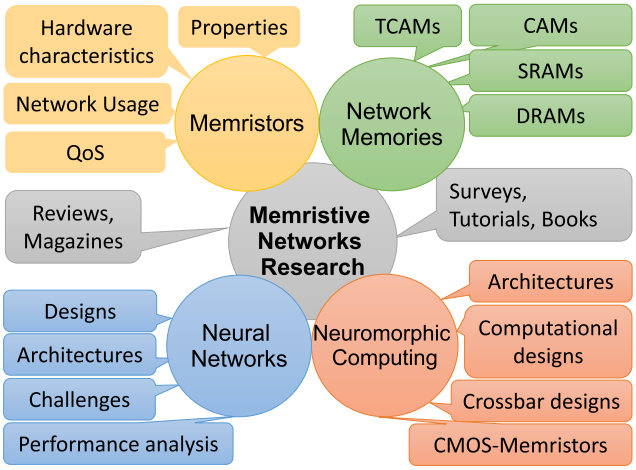

FIGURE 15. Domains of memristor related research in comparison to our survey paper.

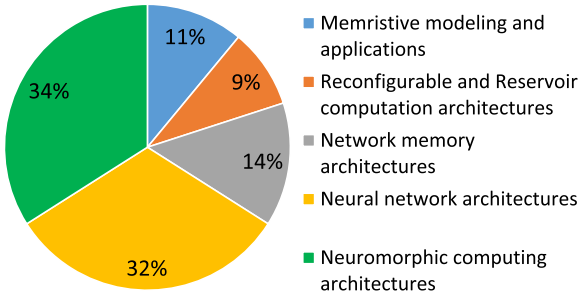

FIGURE 16. Classification of prior surveys.

and advantages of moving towards resistive architectures. The study suggested that slow-down of Moore's law and less improvement in computer architectures is a signal of moving towards new base designs for computer architectures. Also, comparison of previous studies shows that resistive switching can co-localize computing and memory capabilities which provides enhanced benefits in switching speeds, endurance, retention and stacking capabilities. Moreover, emulating the brain's spiking neural capability further promotes resistive switches for development of cognitive functions and data intensive compute applications. Hence, research on novel resistive architectures is one of the prominent lines of future research.

5) COMBINATION OF MEMRISTORS AND TRANSISTORS

Considering the widespread adoption of transistors in network components, a prominent future direction is the hybrid use of transistors and memristors. It also promotes the construction and usage of new components which employ the best properties of both components. In this regard, Talsma et al. [136] proposed the use of memtransistor (or synaptic transistors) for development of ANNs. The authors highlighted that the proposed Single walled carbon nanotubes (SWCNT) transistor exhibits spike timing dependent plasticity like the neurons in brain, which aids in human like learning behavior. The study showed through simulations

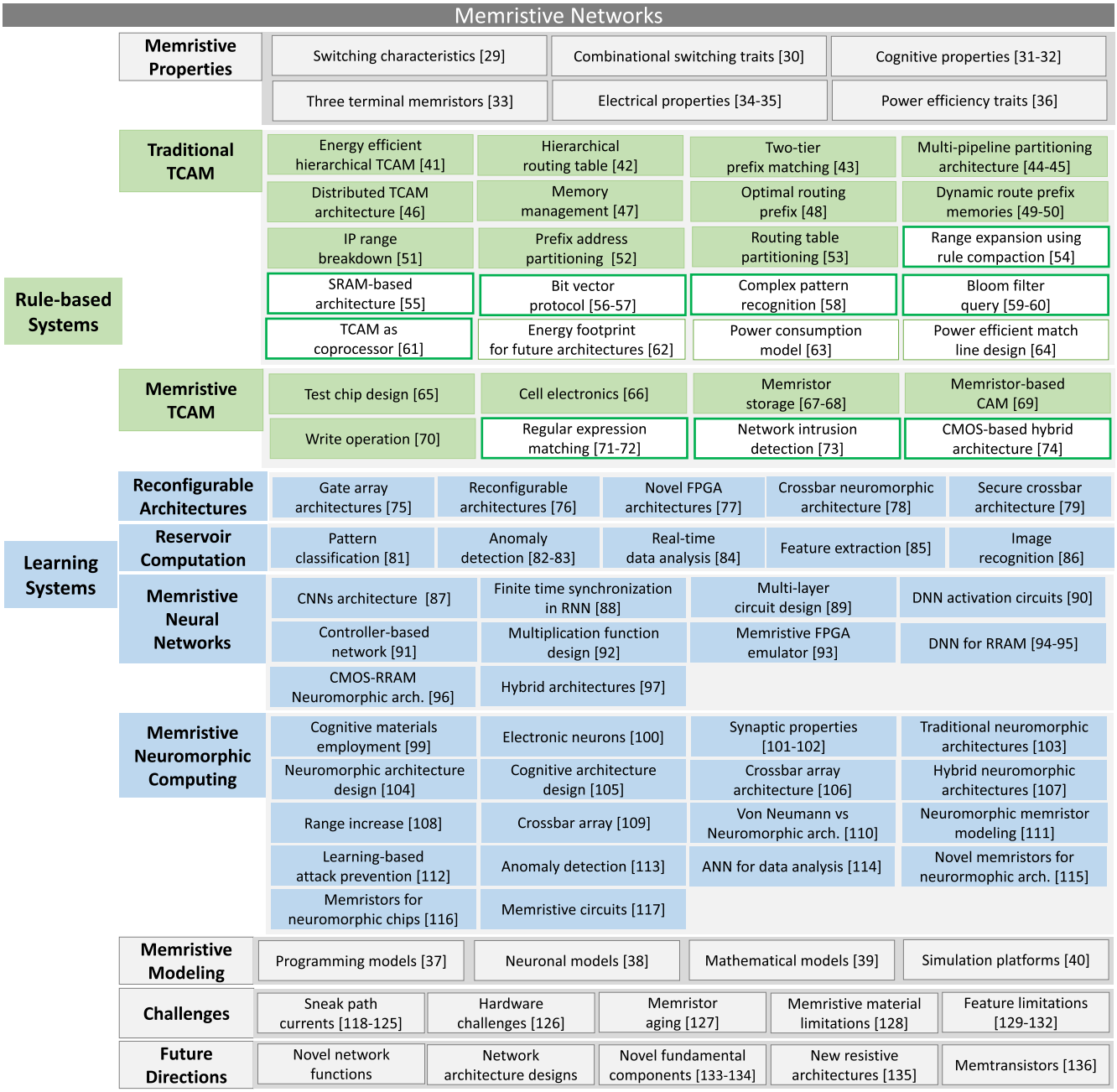

FIGURE 17. Subset of memristor-based researches referred in our survey paper.

that the proposed memtransistor-based ANN exhibits anti-hebbian learning (like human brain) and plasticity effects (like brain neurons). However, more research on a wide array of novel materials with new features is required to explore the use of memtransistors in packet processors.

VII. RELATED WORK

In this section, we review the relevant related researches in order to study the advent of the cognitive network architectures. Fig. 15 presents an overview of various dimensions in related literature catered by the previous researches. The classification of related work in form of a pie chart is shown in Fig. 16 and it shows that surveys on neuromorphic computing gained a lot of attention by the research community.

A. RULE-BASED SYSTEMS USING TCAM

The amount of research in the domain of network memories has always been limited considering the challenging scope of hardware implementation, requirement of physical resources and necessity of a wide range of expertise. However, some researchers explored the network memories

TABLE 3. Summary of existing surveys and reviews related to memristive modeling, network memories and computational architectures.

Related surveys	Topics	Key contributions	Enhancements in our paper
Memristive modeling and applications			
Krestinskaya et al. [144]	Hierarchical temporal memory using memristors	A survey on analog and digital human cortex inspired designs for on-chip near sensor data processing using memristors	Our survey extends the scope of this research to network memories and cognitive packet processors using memristors
Arafin et al. [145]	Memristor modeling and security applications	A survey on memristor abstraction models and implications on hardware security primitives including encryption and storage	Our survey extends the scope of this research for the realization of memristor-based memories, neural networks and neuromorphic architectures
Marani et al. [159]	Memristor applications	A survey reviewing the practicality of memristors in terms of their applications and identification of the potential upper bounds for maximizing memristors' benefits	Our survey extends the scope of this research for novel network memories, neural networks and neuromorphic computing with more up-to-date propositions
Network memory architectures			
Pagiamtzis and Sheikholeslami [137]	CAM circuits and architectures	A survey on the developed CAM circuits and architectures by various researches. It also focuses on the power consumption for the search and match lines, and associated CAM components	Our survey extends the scope of this review to the most recent researches of TCAM memories with an enhanced focus on cognitive materials and architectures
Memristive computational architectures			
Woods et al. [146]	Device modeling using memristor-based crossbar architectures	A survey on the realization of memristive crossbar architectures for device modeling and identification of loopholes in the memristor-based models along with the pathways to optimal memristor models	Our survey extends the scope of this research to the challenges in crossbar architectures and mapping memristor-based models for realization of network memories, neural networks and neuromorphic architectures
Vourkas and Sirakoulis [93]	Memristor-based logic circuit designs	A review focusing on the memristor-based logic circuit designs for network applications and presents a comprehensive layout for circuit designs	Our survey extends the scope of this study to the design of network memories, neural networks and neuromorphic hardware
Maan et al. [160]	Memristive threshold logic circuits	A survey of Memristive Threshold Logic (MTL) circuits inspired by the brain and focuses on the potential applications along with learning boundaries for memristors	We extend the scope of the survey to memristor-based neuromorphic architectures, neural networks and network memories.

especially CAMs, in order to survey the available circuits and architectures [137]. A glimpse of the relevant research and comparison to our survey paper has been presented in Table 3. Here, we would like to highlight some surveys reviewing network memories, but, they have not been mentioned in Table 3 because scope of their study is not directly related to our survey paper. In this context, Taylor [138] presented a thorough taxonomy of packet classification which is directly used by network memories including CAMs and TCAMs. Their survey showcased the routing table implementation strategies for TCAM and laid the foundation for future research. In another research, Sakellari [139] reviewed the cognitive packet networks in context to their Quality of Service (QoS) requirements for multimedia applications. Contrary to our survey paper focusing on memristor-based TCAM memories, Bjerregaard and Mahadevan [140] explore the network on chip architectures. The survey reviews the studies for moving from computation-centric to communication-centric architectures and implementation of scalable communication infrastructures. Our survey complements their efforts in further strengthening the domain of cognitive network memory architectures.

B. NOVEL COMPUTING ARCHITECTURES USING NEW MATERIALS

Historically, pace for the development of novel components using new materials slowed down with time due to a shift in focus from hardware side to the application side of network designs [141], [142], [143]. In this context, the research on memristors was conducted theoretically and computational architectures focused on the software side by using the network resources more efficiently. Several studies [144], [145], [146] presented the theoretical formulations, properties characterization and coupling of memristors with other components. Some surveys in the domain of memristor applications for novel components have been conducted in the recent past with a special focus on memristor modeling, characterization, Hierarchical Temporal Memory (HTM) applications and crossbar array architectures, as shown in Table 3. However, scope of these surveys does not span over memristor-based network memories and cognitive packet processors.

C. NEURAL NETWORKS

Since the development of neural networks in 1944 by Warren McCullough and Walter Pitts, neural networks have fascinated the research community owing to their potential of solving complex pattern recognition problems [147]. A number of studies have surveyed and reviewed neural networks including [148] with a special focus on automatic target recognition, [149] emphasizing the fuzzy neural networks and [150] showcasing neural networks for complex problems including traveling salesman problem. In recent past, neural network surveys have showed great performance

TABLE 4. Summary of existing surveys, reviews and tutorials related to neural networks and neuromorphic computing architectures.

Related surveys	Topics	Key contributions	Enhancements in our paper
Neural network designs and architectures			
Liu et al. [104]	DNN architectures	A survey of deep learning architectures and neural network designs. It presents the applications of deep learning in the domains of speech processing, pattern recognition and computer vision	Our survey enhances this review to a specific focus on memristor-based neural networks and focuses on the application side of networks using cognitive materials
Abiodun et al. [161]	ANN applications	A survey presenting the taxonomy of neural network applications in real world scenarios including case studies, challenges, performance comparisons and shortcomings	Our survey enhances the scope of this research to the application of cognitive materials in network memories, neural networks, and neuromorphic computing
Marugán et al. [162]	ANN for wind energy systems	A survey on the current and future ANN techniques for wind energy systems covering prediction, design optimization, fault detection and optimal control	Our survey spans from ANNs to cognitive architectures especially in the perspective of energy efficient memristors inside packet processors
Qin et al. [163]	Convolutional neural network visualizations	A survey on the visualization techniques for the CNN with a focus on neural network motivations, algorithms, experimental results, and possible implementations	Our survey enhances the neural network review with an implementation perspective considering the memristors and our study expands the scope to network memories and neuromorphic architectures
Neuromorphic computing architectures			
Schuman et al. [164]	Neuromorphic computing and neural networks in hardware	A review on neuromorphic computing and its potential applications along with the possible challenges. The review also explores the software, hardware and application perspective of neuromorphic research	Our survey focuses on the practical realizations of neuromorphic architectures using memristors. Moreover, we present the use of memristors in energy efficient network computation and storage architectures
James et al. [165]	Software and hardware aspects of neuromorphic computing	A review on the algorithms leveraging from the fundamentals of biological organisms to computational hardware. It also surveys the co-localization of memory inside the neuromorphic architecture by research on novel materials	Our survey pinpoints the use of memristors for colocalization aspects of neuromorphic computing and neural networks with an up-to-date literature review
Sung et al. [166]	Memristive hardware for neuromorphic computation	A review on the competitiveness and challenges of memristors for neuromorphic computation in comparison to CMOS-based Graphics Processing Units (GPUs), FPGAs and ASICs	Our survey focuses on an indepth analysis of memristor-based neuromorphic architectures and covers the hardware aspects along with the challenges and future directions
Burr et al. [167]	Neuromorphic computing using non-volatile memories	A survey on application of Non-Volatile Memory (NVM) for SNN, DNN and Mem-computing and review of algorithms for neuromorphic computing	Our survey enhances memristor-based computing to neural networks and considers the problem with an application perspective
Davies et al. [168]	Neuromorphic computing with Loihi	A survey of neuromorphic computing benchmarks on *Loihi* chip with state-of-the-art conventional approaches	Our paper extends the scope of neuromorphic computing and neural networks to memristors for practical realizations
Saxena [169]	Neuromorphic devices and integrated circuits	A review of neuromorphic integrated circuits with a hybrid CMOS-RRAM implementation using SNN and practical device considerations	Our survey enhances the study for neuromorphic architectures with memristor-based practical applications
Volanis et al. [170]	Cognitive neuromorphic ICs	A tutorial on neuromorphic circuits, synapses, learning algorithms, information transmission, neural networks and available platforms	Our paper enhances the scope of this tutorial to a survey of researches on neuromorphic computing and neural networks with an application perspective by using memristors as cognitive components

for resource constraint applications [151], natural language processing [152] and application-specific research problems like breast cancer detection [153]. However, the scope of these surveys is more diverse and lacks an implementation perspective in context to networks with a focus on cognitive materials. Table 4 presents the relevant neural network surveys with findings relevant to the field of networks. A comparison of studies shows that our survey highlights and compares a thorough implementation perspective of memristor-based neural networks which has not been presented before.

D. NEUROMORPHIC COMPUTING

In comparison to multiple sub-domains in our survey paper, most of the relevant related literature surveys exist in the area of neuromorphic computing. Table 4 presents their contributions in comparison to our survey paper. In this subsection, we would like to showcase some neuromorphic studies presenting some major findings but not directly related to our survey paper. In this regard, Yang and Kim [154] present the Intel's neuromorphic chip design and compare it with other neuromorphic chips for emulating brain functioning. Olin-Ammentorp and Cady [155] reviewed the neuromorphic

computing research and presented the associated neural networks and architectures for practical implementations. Li et al. [156] completed a thorough review of memristor usage in neuromorphic devices. The major focus of all these researches was the study of hardware challenges and material properties. Nawrocki et al. [157] present a review on the available neuromorphic architectures and systems along with their implementation strategies. Rajendran and Alibart [158] reviewed the neuromorphic architectures in the light of emerging memory devices and compared the performance with the brain's cognitive functioning. However, comparison of all previous surveys and reviews showed that our survey presents the fundamental concepts in understanding the implementation of cognitive packet processors.

E. OUR SURVEY AND CONTRIBUTIONS

Our survey plays a fundamental role in extending the research on networks from an implementation perspective. Moreover, our survey extends the cognitive view of networks for the current performance and resource constraint networks. Fig. 17 shows the taxonomy of major researches regarding memristor-based architectures referred in our survey paper. The major contributions of our survey paper in comparison to previous surveys and reviews are summarized in below lines;

- *Memristor-based packet processors:* Our survey presents a thorough technical review for using memristors in the design of cognitive packet processors with a *Networks* perspective. We present the memristive analysis for the computer science and computer networks domains.
- *Rule-based systems using memristors:* We study the use of memristors for rule-based systems using TCAM architectures. We show the state-of-the-art architectural configurations and challenging power requirements for TCAM. We present a thorough review of all researches presenting a nanoscale implementation of memristors inside TCAM.
- *Memristor-based Learning systems:* We highlight the use of memristors for computational tasks based upon the learning systems. We discuss the design of reconfigurable architectures and reservoir computation architectures. We present an elaborate review on the use of memristors for neural networks in reference to the related research. We present a detailed analysis on memristor-based neuromorphic computing. We compare the previous state-of-the-art approaches and present findings for future researches.

Comparison of our survey with previous surveys shows that our survey plays a fundamental role in understanding memristor-based architectures for future cognitive networks. We also explore the challenges and future directions of research for memristor-based architectures. Moreover, the key insights in our survey paper open the horizon for future research in the domain of novel cognitive materials for use in future packet processors.

VIII. CONCLUSION

This survey reviews the memristor-based cognitive computational architectures for energy efficiency and cognitive functionality in network packet processors. Research shows that memristors provide better read and write properties in form of less power consumption and stateful operations. The programmability of memristors can be used for deployment of complex network functions in rule-based systems and learning systems at packet processors. In rule-based systems, memristors can be used for cognitive information handling through TCAM architectures. The non-volatile and stateful response of memristors can be used to store and search entries in one clock cycle, but, with less energy and area consumption due to the memristive atomic operations and nanoscale size. Learning systems can benefit from the programmability of memristors through crossbar array architectures and reservoir computation architectures. The computational components can be extended to neural network architectures by connecting memristive network in form of an array of neurons with programmable memristive conductances analogous to the neural weights. Moreover, removing synchronous clock in combination with the model of spiking neural networks can aid in developing neuromorphic architectures using memristors. Despite benefits, memristor-based networks face the challenges of sneak path currents, aging effects, combination with CMOS and absence of software protocols to harness from the benefits of memristors. The survey shows that the development of novel network protocols, and network components built upon new network functions utilizing the colocalization of memory and computation are critical factors for future packet processors.

REFERENCES

[1] Intel®. *Tofino™ 3 Intelligent Fabric Processor Brief*. Accessed: Nov. 22, 2022. [Online]. Available: https://www.intel.com/content/www/us/en/products/network-io/programmable-ethernet-switch.html
[2] Office of Nuclear Energy. *How Much Power Does a Nuclear Reactor Produce?*. Accessed: Nov. 22, 2022. [Online]. Available: https://www.energy.gov/ne/articles/infographic-how-much-power-does-nuclear-reactor-produce
[3] IEA. *Data Centres and Data Transmission Networks*. Accessed: Nov. 22, 2022. [Online]. Available: https://www.iea.org/reports/data-centres-and-data-transmission-networks
[4] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, May 2008.
[5] L. Chua, G. C. Sirakoulis, and A. Adamatzky, *Handbook of Memristor Networks*. Cham, Switzerland: Springer, 2019.
[6] Z. Xiong and N. Zilberman, "Do switches dream of machine learning? Toward in-network classification," in *Proc. 18th ACM Workshop Hot Topics Netw.*, Nov. 2019, pp. 25–33.
[7] L. Huo, D. Jiang, S. Qi, H. Song, and L. Miao, "An AI-based adaptive cognitive modeling and measurement method of network traffic for EIS," *Mobile Netw. Appl.*, vol. 26, no. 2, pp. 575–585, Apr. 2021.
[8] O. Popoola and B. Pranggono, "On energy consumption of switch-centric data center networks," *J. Supercomput.*, vol. 74, no. 1, pp. 334–369, Jan. 2018.
[9] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: An intellectual history of programmable networks," *Queue*, vol. 11, no. 12, pp. 20–40, Dec. 2013.
[10] J. Wang, *Handbook of Finite State Based Models and Applications*. Boca Raton, FL, USA: CRC Press, 2012.

[11] K. O'Neal and P. Brisk, "Predictive modeling for CPU, GPU, and FPGA performance and power consumption: A survey," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Jul. 2018, pp. 763–768.

[12] The World Bank. (2021). *World Development Report 2021: Data for Better Lives*. Accessed: Nov. 22, 2022. [Online]. Available: https://www.worldbank.org/en/publication/wdr2021

[13] C. L. Hedrick, *Routing Information Protocol*, document RFC 1058, RFC Editor, Jun. 1988, p. 33. [Online]. Available: https://www.rfc-editor.org/info/rfc1058, doi: 10.17487/RFC1058.

[14] M. Latah and L. Toker, "Artificial intelligence enabled software-defined networking: A comprehensive overview," *IET Netw.*, vol. 8, no. 2, pp. 79–99, Mar. 2019.

[15] A. Ligeza, *Logical Foundations for Rule-Based Systems*, vol. 11. Heidelberg, Germany: Springer, 2006.

[16] B. G. Buchanan, T. M. Mitchell, R. G. Smith, and C. R. Johnson, Jr., "Models of learning systems," in *Encylopedia of Computer Science and Technology*, vol. 11, J. Belzer, A. G. Holzman, and A. Kent, Eds. New York, NY, USA: Marcel Dekker, 1977, pp. 24–51. [Online]. Available: https://www.cs.cmu.edu/~tom/pubs/Models_of_Learning_Systems_1978.pdf

[17] L. Chisvin and R. J. Duckworth, "Content-addressable and associative memory: Alternatives to the ubiquitous RAM," *Computer*, vol. 22, no. 7, pp. 51–64, Jul. 1989.

[18] P. Bosshart, D. Daly, G. Gibb, M. Izzard, N. McKeown, J. Rexford, C. Schlesinger, D. Talayco, A. Vahdat, G. Varghese, and D. Walker, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 87–95, 2014.

[19] R. Singh, M. Mukhtar, A. Krishna, A. Parkhi, J. Padhye, and D. Maltz, "Surviving switch failures in cloud datacenters," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 51, no. 2, pp. 2–9, Apr. 2021.

[20] Y. Chu, *High-Level Language Computer Architecture*. New York, NY, USA: Academic, 2014.

[21] *Rebooting the IT Revolution: A Call to Action*, Semiconductor Industry Association, Washington, DC, USA, 2015.

[22] J. Von Neumann and R. Kurzweil, *The Computer and the Brain*. New Haven, CT, USA: Yale Univ. Press, 2012.

[23] M. Davies et al., "Loihi: A neuromorphic many core processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, Feb. 2018.

[24] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, and N. Imam, "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 34, no. 10, pp. 1537–1557, Oct. 2015.

[25] L. O. Chua, "Memristor—The missing circuit element," *IEEE Trans. Circuit Theory*, vol. CT-18, no. 5, pp. 507–519, Sep. 1971.

[26] L. O. Chua and S. M. Kang, "Memristive devices and systems," *Proc. IEEE*, vol. 64, no. 2, pp. 209–223, Feb. 1976.

[27] J. J. Yang, M. D. Pickett, X. Li, D. A. A. Ohlberg, D. R. Stewart, and R. S. Williams, "Memristive switching mechanism for metal/oxide/metal nanodevices," *Nature Nanotechnol.*, vol. 3, no. 7, pp. 429–433, 2008.

[28] M. Khalid, "Review on various memristor models, characteristics, potential applications, and future works," *Trans. Electr. Electron. Mater.*, vol. 20, no. 4, pp. 289–298, Aug. 2019.

[29] T. Prodromakis, K. Michelakis, and C. Toumazou, "Switching mechanisms in microscale memristors," *Electron. Lett.*, vol. 46, no. 1, pp. 63–65, 2010.

[30] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: Properties of basic electrical circuits," *Eur. J. Phys.*, vol. 30, no. 4, p. 661, 2009.

[31] P. Sheridan and W. Lu, "Memristors and memristive devices for neuromorphic computing," in *Handbook of Memristor Networks*. Cham, Switzerland: Springer, 2019, pp. 369–389.

[32] P. Sheridan and W. Lu, "Memristors and memristive devices for neuromorphic computing," in *Memristor Networks*. Cham, Switzerland: Springer, 2014, pp. 129–149.

[33] Q. Duan, T. Zhang, M. Yin, C. Cheng, L. Xu, Y. Yang, and R. Huang, "Switching dynamics and computing applications of memristors: An overview," in *Proc. IEEE 12th Int. Conf. ASIC (ASICON)*, Oct. 2017, pp. 144–147.

[34] Y. Ho, G. M. Huang, and P. Li, "Nonvolatile memristor memory: Device characteristics and design implications," in *Proc. Int. Conf. Comput.-Aided Design (ICCAD)*, 2009, pp. 485–490.

[35] Y. Ho, G. M. Huang, and P. Li, "Dynamical properties and design analysis for nonvolatile memristor memories," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 58, no. 4, pp. 724–736, Apr. 2011.

[36] A. G. Radwan and M. E. Fouda, *On the Mathematical Modeling of Memristor, Memcapacitor, and Meminductor*, vol. 26. Berlin, Germany: Springer, 2015.

[37] A. Bala, A. Adeyemo, X. Yang, and A. Jabir, "High level abstraction of memristor model for neural network simulation," in *Proc. 6th Int. Symp. Embedded Comput. Syst. Design (ISED)*, Dec. 2016, pp. 318–322.

[38] D. Zhevnenko, F. Meshchaninov, V. Kozhevnikov, E. Shamin, A. Belov, S. Gerasimova, D. Guseinov, A. Mikhaylov, and E. Gornev, "Simulation of memristor switching time series in response to spike-like signal," *Chaos, Solitons Fractals*, vol. 142, Jan. 2021, Art. no. 110382.

[39] H. Lin, C. Wang, Q. Hong, and Y. Sun, "A multi-stable memristor and its application in a neural network," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 67, no. 12, pp. 3472–3476, Dec. 2020.

[40] L. Xia, B. Li, T. Tang, P. Gu, P.-Y. Chen, S. Yu, Y. Cao, Y. Wang, Y. Xie, and H. Yang, "MNSIM: Simulation platform for memristor-based neuromorphic computing system," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 5, pp. 1009–1022, Jul. 2017.

[41] E. Spitznagel, D. Taylor, and J. Turner, "Packet classification using extended TCAMs," in *Proc. 11th IEEE Int. Conf. Netw. Protocols*, Oct. 2003, pp. 120–131.

[42] V. C. Ravikumar and R. N. Mahapatra, "TCAM architecture for IP lookup using prefix properties," *IEEE Micro*, vol. 24, no. 2, pp. 60–69, Mar. 2004.

[43] S. Kasnavi, V. C. Gaudet, P. Berube, and J. N. Amaral, "A hardware-based longest prefix matching scheme for TCAMs," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2005, pp. 3339–3342.

[44] W. Jiang and V. K. Prasanna, "Energy-efficient multi-pipeline architecture for terabit packet classification," in *Proc. IEEE Global Telecommun. Conf.*, Nov. 2009, pp. 1–6.

[45] W. Jiang, Q. Wang, and V. K. Prasanna, "Beyond TCAMs: An SRAM-based parallel multi-pipeline architecture for terabit IP lookup," in *Proc. 27th Conf. Comput. Commun.*, Apr. 2008, pp. 1786–1794.

[46] K. Zheng, C. Hu, H. Lu, and B. Liu, "A TCAM-based distributed parallel IP lookup scheme and performance analysis," *IEEE/ACM Trans. Netw.*, vol. 14, no. 4, pp. 863–875, Aug. 2006.

[47] T. Mishra and S. Sahni, "DUOS—Simple dual TCAM architecture for routing tables with incremental update," in *Proc. IEEE Symp. Comput. Commun.*, Jun. 2010, pp. 503–508.

[48] T. Mishra and S. Sahni, "PETCAM—A power efficient TCAM architecture for forwarding tables," *IEEE Trans. Comput.*, vol. 61, no. 1, pp. 3–17, Jan. 2012.

[49] H. Liu, "Routing table compaction in ternary CAM," *IEEE Micro*, vol. 22, no. 1, pp. 58–64, Jan. 2002.

[50] M. J. Akhbarizadeh, M. Nourani, R. Panigrahy, and S. Sharma, "A TCAM-based parallel architecture for high-speed packet forwarding," *IEEE Trans. Comput.*, vol. 56, no. 1, pp. 58–72, Jan. 2007.

[51] Y.-K. Chang, "A 2-level TCAM architecture for ranges," *IEEE Trans. Comput.*, vol. 55, no. 12, pp. 1614–1629, Dec. 2006.

[52] R. Panigrahy and S. Sharma, "Reducing TCAM power consumption and increasing throughput," in *Proc. 10th Symp. High Perform. Interconnects*, 2002, pp. 107–112.

[53] F. Zane, G. Narlikar, and A. Basu, "CoolCAMs: Power-efficient TCAMs for forwarding engines," in *Proc. 22nd Annu. Joint Conf. IEEE Comput. Commun. Societies*, vol. 1, Apr. 2003, pp. 42–52.

[54] A. X. Liu, C. R. Meiners, and E. Torng, "TCAM razor: A systematic approach towards minimizing packet classifiers in TCAMs," *IEEE/ACM Trans. Netw.*, vol. 18, no. 2, pp. 490–500, Apr. 2010.

[55] Z. Ullah, M. K. Jaiswal, and R. C. C. Cheung, "Z-TCAM: An SRAM-based architecture for TCAM," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 2, pp. 402–406, Feb. 2015.

[56] F. Baboescu and G. Varghese, "Scalable packet classification," *IEEE/ACM Trans. Netw.*, vol. 13, no. 1, pp. 2–14, Feb. 2005.

[57] F. Baboescu and G. Varghese, "Scalable packet classification," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 4, pp. 199–210, 2001.

[58] F. Yu, R. H. Katz, and T. V. Lakshman, "Gigabit rate packet pattern-matching using TCAM," in *Proc. 12th IEEE Int. Conf. Netw. Protocols*, Oct. 2004, pp. 174–183.

[59] S. Dharmapurikar, P. Krishnamurthy, and D. E. Taylor, "Longest prefix matching using Bloom filters," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2003, pp. 201–212.

[60] S. Dharmapurikar, P. Krishnamurthy, and D. E. Taylor, "Longest prefix matching using Bloom filters," *IEEE/ACM Trans. Netw.*, vol. 14, no. 2, pp. 397–409, Apr. 2006.

[61] S. Ahmad and R. Mahapatra, "TCAM enabled on-chip logic minimization," in *Proc. 42nd Design Autom. Conf.*, 2005, pp. 678–683.

[62] C. Chen, D. Barrera, and A. Perrig, "Modeling data-plane power consumption of future internet architectures," in *Proc. IEEE 2nd Int. Conf. Collaboration Internet Comput. (CIC)*, Nov. 2016, pp. 149–158.
[63] B. Agrawal and T. Sherwood, "Modeling TCAM power for next generation network devices," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw.*, Mar. 2006, pp. 120–129.
[64] K. Mathan and T. Ravichandran, "Data intelligent low power high performance TCAM for IP-address lookup table," *Circuits Syst.*, vol. 7, no. 11, p. 3734, 2016.
[65] I. Arsovski, T. Chandler, and A. Sheikholeslami, "A ternary content-addressable memory (TCAM) based on 4T static storage and including a current-race sensing scheme," *IEEE J. Solid-State Circuits*, vol. 38, no. 1, pp. 155–158, Jan. 2003.
[66] Q. Guo, X. Guo, Y. Bai, and E. Ipek, "A resistive TCAM accelerator for data-intensive computing," in *Proc. 44th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Dec. 2011, pp. 339–350.
[67] P. Junsangsri and F. Lombardi, "A memristor-based TCAM (ternary content addressable memory) cell: Design and evaluation," in *Proc. Great Lakes Symp. VLSI*, 2012, pp. 311–314.
[68] P. Junsangsri, F. Lombardi, and J. Han, "A memristor-based TCAM (ternary content addressable memory) cell," in *Proc. IEEE/ACM Int. Symp. Nanosc. Archit. (NANOARCH)*, Jul. 2014, pp. 1–6.
[69] W. Chen, X. Yang, and F. Z. Wang, "Memristor content addressable memory," in *Proc. IEEE/ACM Int. Symp. Nanosc. Archit. (NANOARCH)*, Jul. 2014, pp. 83–87.
[70] A. G. Ruotolo, M. Ottavi, S. Pontarelli, and F. Lombardi, "A novel write-scheme for data integrity in memristor-based crossbar memories," in *Proc. IEEE/ACM Int. Symp. Nanosc. Archit.*, Jul. 2012, pp. 168–173.
[71] C. E. Graves, W. Ma, X. Sheng, B. Buchanan, L. Zheng, S.-T. Lam, X. Li, S. R. Chalamalasetti, L. Kiyama, M. Foltin, M. P. Hardy, and J. P. Strachan, "Regular expression matching with memristor TCAMs," in *Proc. IEEE Int. Conf. Rebooting Comput. (ICRC)*, Nov. 2018, pp. 1–11.
[72] C. E. Graves, W. Ma, X. Sheng, B. Buchanan, L. Zheng, S.-T. Lam, X. Li, S. R. Chalamalasetti, L. Kiyama, M. Foltin, J. P. Strachan, and M. P. Hardy, "Regular expression matching with memristor TCAMs for network security," in *Proc. 14th IEEE/ACM Int. Symp. Nanosc. Archit.*, Jul. 2018, pp. 65–71.
[73] C. E. Graves, C. Li, X. Sheng, W. Ma, S. R. Chalamalasetti, D. Miller, J. S. Ignowski, B. Buchanan, L. Zheng, S.-T. Lam, X. Li, L. Kiyama, M. Foltin, M. P. Hardy, and J. P. Strachan, "Memristor TCAMs accelerate regular expression matching for network intrusion detection," *IEEE Trans. Nanotechnol.*, vol. 18, pp. 963–970, 2019.
[74] S. Tabassum, F. Parveen, and A. B. M. Harun-Rashid, "Low power high speed ternary content addressable memory design using MOSFET and memristors," in *Proc. Int. Conf. Electron. Commun. Syst. (ICECS)*, Feb. 2014, pp. 1–6.
[75] J. Rajendran, H. Manem, R. Karri, and G. S. Rose, "Memristor based programmable threshold logic array," in *Proc. IEEE/ACM Int. Symp. Nanosc. Archit.*, Jun. 2010, pp. 5–10.
[76] W. Lu, K.-H. Kim, T. Chang, and S. Gaba, "Two-terminal resistive switches (memristors) for memory and logic applications," in *Proc. 16th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2011, pp. 217–223.
[77] J. Cong and B. Xiao, "MrFPGA: A novel FPGA architecture with memristor-based reconfiguration," in *Proc. IEEE/ACM Int. Symp. Nanosc. Archit.*, Jun. 2011, pp. 1–8.
[78] V. Bontupalli, C. Yakopcic, R. Hasan, and T. M. Taha, "Efficient memristor-based architecture for intrusion detection and high-speed packet classification," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 14, no. 4, pp. 1–27, Dec. 2018.
[79] S. Kannan, N. Karimi, and O. Sinanoglu, "Secure memristor-based main memory," in *Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.
[80] R. Ricart-Sanchez, P. Malagon, J. M. Alcaraz-Calero, and Q. Wang, "P4-NetFPGA-based network slicing solution for 5G MEC architectures," in *Proc. ACM/IEEE Symp. Archit. Netw. Commun. Syst. (ANCS)*, Sep. 2019, pp. 1–2.
[81] M. S. Kulkarni and C. Teuscher, "Memristor-based reservoir computing," in *Proc. IEEE/ACM Int. Symp. Nanosc. Archit.*, Jul. 2012, pp. 226–232.
[82] S. Sayyaparaju, M. S. A. Shawkat, M. M. Adnan, and G. S. Rose, "Circuit techniques for efficient implementation of memristor based reservoir computing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Oct. 2020, pp. 1–5.
[83] S. Sayyaparaju, M. S. Ara Shawkat, and G. S. Rose, "Robust implementation of memristive reservoir computing with crossbar based readout layer," in *Proc. IEEE 14th Dallas Circuits Syst. Conf. (DCAS)*, Nov. 2020, pp. 1–4.
[84] X. Zhu, Q. Wang, and W. D. Lu, "Memristor networks for real-time neural activity analysis," *Nature Commun.*, vol. 11, no. 1, pp. 1–9, Dec. 2020.
[85] F. Cai and W. D. Lu, "Feature extraction and analysis using memristor networks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2018, pp. 1–4.
[86] H. Ran, S. Wen, Q. Li, Y. Yang, K. Shi, Y. Feng, P. Zhou, and T. Huang, "Memristor-based edge computing of blaze block for image recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 5, pp. 2121–2131, May 2022.
[87] E. Lehtonen and M. Laiho, "CNN using memristors for neighborhood connections," in *Proc. 12th Int. Workshop Cellular Nanoscale Netw. Appl. (CNNA)*, Feb. 2010, pp. 1–4.
[88] J. Gao, P. Zhu, A. Alsaedi, F. E. Alsaadi, and T. Hayat, "A new switching control for finite-time synchronization of memristor-based recurrent neural networks," *Neural Netw.*, vol. 86, pp. 1–9, Feb. 2017.
[89] Y. Zhang, X. Wang, and E. G. Friedman, "Memristor-based circuit design for multilayer neural networks," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 2, pp. 677–686, Feb. 2017.
[90] A. Bala, X. Yang, A. Adeyemo, and A. Jabir, "A memristive activation circuit for deep learning neural networks," in *Proc. 8th Int. Symp. Embedded Comput. Syst. Design (ISED)*, Dec. 2018, pp. 1–5.
[91] A. Wu and Z. Zeng, "Exponential stabilization of memristive neural networks with time delays," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 12, pp. 1919–1929, Dec. 2012.
[92] S. Duan, X. Hu, Z. Dong, L. Wang, and P. Mazumder, "Memristor-based cellular nonlinear/neural network: Design, analysis, and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1202–1213, Jun. 2015.
[93] I. Vourkas, A. Abusleme, V. Ntinas, G. C. Sirakoulis, and A. Rubio, "A digital memristor emulator for FPGA-based artificial neural networks," in *Proc. 1st IEEE Int. Verification Secur. Workshop (IVSW)*, Jul. 2016, pp. 1–4.
[94] Q. Wang, X. Wang, S. H. Lee, F.-H. Meng, and W. D. Lu, "A deep neural network accelerator based on tiled RRAM architecture," in *IEDM Tech. Dig.*, Dec. 2019, pp. 4–14.
[95] X. Wang, Q. Wang, F.-H. Meng, S. H. Lee, and W. D. Lu, "Deep neural network mapping and performance analysis on tiled RRAM architecture," in *Proc. 2nd IEEE Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Aug. 2020, pp. 141–144.
[96] J. M. Correll, V. Bothra, F. Cai, Y. Lim, S. H. Lee, S. Lee, W. D. Lu, Z. Zhang, and M. P. Flynn, "A fully integrated reprogrammable CMOS-RRAM compute-in-memory coprocessor for neuromorphic applications," *IEEE J. Explor. Solid-State Comput. Devices Circuits*, vol. 6, pp. 36–44, 2020.
[97] K. Van Pham and K.-S. Min, "Non-ideal effects of memristor-CMOS hybrid circuits for realizing multiple-layer neural networks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.
[98] D. Marković, A. Mizrahi, D. Querlioz, and J. Grollier, "Physics for neuromorphic computing," *Nat. Rev. Phys.*, vol. 2, pp. 499–510, Jul. 2020.
[99] W.-H. Chen, W.-S. Khwa, J.-Y. Li, W.-Y. Lin, H.-T. Lin, Y. Liu, Y. Wang, H. Wu, H. Yang, and M.-F. Chang, "Circuit design for beyond von Neumann applications using emerging memory: From nonvolatile logics to neuromorphic computing," in *Proc. 18th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2017, pp. 23–28.
[100] W. Yi, K. K. Tsang, S. K. Lam, X. Bai, J. A. Crowell, and E. A. Flores, "Active memristor neurons for neuromorphic computing," in *IEDM Tech. Dig.*, Dec. 2019, pp. 7–22.
[101] N. K. Upadhyay, S. Joshi, and J. J. Yang, "Synaptic electronics and neuromorphic computing," *Sci. China Inf. Sci.*, vol. 59, no. 6, pp. 1–26, Jun. 2016.
[102] X. Zhang, A. Huang, Q. Hu, Z. Xiao, and P. K. Chu, "Neuromorphic computing with memristor crossbar," *Phys. Status Solidi A*, vol. 215, no. 13, Jul. 2018, Art. no. 1700875.
[103] G. Indiveri and S. Liu, "Memory and information processing in neuromorphic systems," *Proc. IEEE*, vol. 103, no. 8, pp. 1379–1397, Jan. 2015.
[104] C. Liu, F. Liu, and H. Li, "Brain-inspired computing accelerated by memristor technology," in *Proc. 4th ACM Int. Conf. Nanosc. Comput. Commun.*, Sep. 2017, pp. 1–6.

[105] L. Zheng, S. Shin, and S.-M.-S. Kang, "Memristor-based synapses and neurons for neuromorphic computing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 1150–1153.

[106] M. Hu, H. Li, Y. Chen, Q. Wu, G. S. Rose, and R. W. Linderman, "Memristor crossbar-based neuromorphic computing system: A case study," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1864–1878, Oct. 2014.

[107] X. Liu, M. Mao, B. Liu, B. Li, Y. Wang, H. Jiang, and M. Barnell, "Harmonica: A framework of heterogeneous computing systems with memristor-based neuromorphic computing accelerators," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 5, pp. 617–628, May 2016.

[108] H. Jiang, W. Zhu, F. Luo, K. Bai, C. Liu, X. Zhang, J. J. Yang, Q. Xia, Y. Chen, and Q. Wu, "Cyclical sensing integrate-and-fire circuit for memristor array based neuromorphic computing," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 930–933.

[109] I. Kataeva, S. Ohtsuka, H. Nili, H. Kim, Y. Isobe, K. Yako, and D. Strukov, "Towards the development of analog neuromorphic chip prototype with 2.4 M integrated memristors," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.

[110] H. An, Z. Zhou, and Y. Yi, "Opportunities and challenges on nanoscale 3D neuromorphic computing system," in *Proc. IEEE Int. Symp. Electromagn. Compat. Signal/Power Integrity (EMCSI)*, Aug. 2017, pp. 416–421.

[111] F. Marrone, G. Zoppo, F. Corinto, and M. Gilli, "Second order memristor models for neuromorphic computing," in *Proc. IEEE 62nd Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2019, pp. 315–318.

[112] C. Yang, B. Liu, H. Li, Y. Chen, M. Barnell, Q. Wu, W. Wen, and J. Rajendran, "Thwarting replication attack against memristor-based neuromorphic computing system," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 39, no. 10, pp. 2192–2205, Oct. 2020.

[113] Q. Chen, Q. Qiu, H. Li, and Q. Wu, "A neuromorphic architecture for anomaly detection in autonomous large-area traffic monitoring," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2013, pp. 202–205.

[114] J. Fu, Z. Liao, and J. Wang, "Memristor-based neuromorphic hardware improvement for privacy-preserving ANN," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 27, no. 12, pp. 2745–2754, Dec. 2019.

[115] A. Zaman, E. Shin, C. Yakopcic, T. M. Taha, and G. Subramanyam, "Experimental study of memristors for use in neuromorphic computing," in *Proc. IEEE Nat. Aerosp. Electron. Conf.*, Jul. 2018, pp. 370–374.

[116] S. Wang, W. Wang, C. Yakopcic, E. Shin, G. Subramanyam, and T. M. Taha, "Experimental study of $LiNbO_3$ memristors for use in neuromorphic computing," *Microelectron. Eng.*, vol. 168, pp. 37–40, Jan. 2017.

[117] C. Zhao, G. Y. Zhou, C. Z. Zhao, L. Yang, K. L. Man, and E. G. Lim, "Memristor-based neuromorphic implementations for artificial neural networks," in *Proc. Int. SoC Design Conf. (ISOCC)*, Nov. 2018, pp. 174–175.

[118] P. O. Vontobel, W. Robinett, P. J. Kuekes, D. R. Stewart, J. Straznicky, and R. S. Williams, "Writing to and reading from a nano-scale crossbar memory based on memristors," *Nanotechnology*, vol. 20, no. 42, Oct. 2009, Art. no. 425204.

[119] H. Manem, G. S. Rose, X. He, and W. Wang, "Design considerations for variation tolerant multilevel CMOS/nano memristor memory," in *Proc. 20th Symp. Great lakes Symp. VLSI*, 2010, pp. 287–292.

[120] W. Fei, H. Yu, W. Zhang, and K. S. Yeo, "Design exploration of hybrid CMOS and memristor circuit by new modified nodal analysis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 6, pp. 1012–1025, Jun. 2012.

[121] S. Kim, H. Y. Jeong, S. K. Kim, S.-Y. Choi, and K. J. Lee, "Flexible memristive memory array on plastic substrates," *Nano Lett.*, vol. 11, no. 12, pp. 5438–5442, 2011.

[122] C.-M. Jung, J.-M. Choi, and K.-S. Min, "Two-step write scheme for reducing sneak-path leakage in complementary memristor array," *IEEE Trans. Nanotechnol.*, vol. 11, no. 3, pp. 611–618, May 2012.

[123] J. J. Yang, M.-X. Zhang, M. D. Pickett, F. Miao, J. P. Strachan, W.-Di Li, and W. Yi, "Engineering nonlinearity into memristors for passive crossbar applications," *Appl. Phys. Lett.*, vol. 100, no. 11, 2012, Art. no. 113501.

[124] M. S. Qureshi, W. Yi, G. Medeiros-Ribeiro, and R. S. Williams, "AC sense technique for memristor crossbar," *Electron. Lett.*, vol. 48, no. 13, pp. 757–758, 2012.

[125] M. A. Zidan, H. A. H. Fahmy, M. M. Hussain, and K. N. Salama, "Memristor-based memory: The sneak paths problem and solutions," *Microelectron. J.*, vol. 44, no. 2, pp. 176–183, 2013.

[126] T. Shin, K. Son, S. Kim, K. Cho, S. Park, S. Kim, G. Park, B. Sim, and J. Kim, "Impact of on-chip interconnection in a large-scale memristor crossbar array for neural network accelerator and neuromorphic chip," in *Proc. IEEE 28th Conf. Electr. Perform. Electron. Packag. Syst. (EPEPS)*, Oct. 2019, pp. 1–3.

[127] S. Zhang, G. L. Zhang, B. Li, H. H. Li, and U. Schlichtmann, "Aging-aware lifetime enhancement for memristor-based neuromorphic computing," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2019, pp. 1751–1756.

[128] R. Wang, T. Shi, X. Zhang, W. Wang, J. Wei, J. Lu, X. Zhao, Z. Wu, R. Cao, S. Long, Q. Liu, and M. Liu, "Bipolar analog memristors as artificial synapses for neuromorphic computing," *Materials*, vol. 11, no. 11, p. 2102, Oct. 2018.

[129] C. Murray, "Basic research needs for microelectronics: Report of the office of science workshop on basic research needs for microelectronics," USDOE Office of Science, Washington, DC, USA, Tech. Rep. 1616249, 2018. [Online]. Available: https://www.osti.gov/biblio/1616249, doi: 10.2172/1616249.

[130] Y. Wang, Z. Zhang, M. Xu, Y. Yang, M. Ma, H. Li, J. Pei, and L. Shi, "Self-doping memristors with equivalently synaptic ion dynamics for neuromorphic computing," *ACS Appl. Mater. Interfaces*, vol. 11, no. 27, pp. 24230–24240, Jul. 2019.

[131] S. Majumdar, H. Tan, Q. H. Qin, and S. van Dijken, "Energy-efficient organic ferroelectric tunnel junction memristors for neuromorphic computing," *Adv. Electron. Mater.*, vol. 5, no. 3, Mar. 2019, Art. no. 1800795.

[132] B. Dang, Q. Wu, F. Song, J. Sun, M. Yang, X. Ma, H. Wang, and Y. Hao, "A bio-inspired physically transient/biodegradable synapse for security neuromorphic computing based on memristors," *Nanoscale*, vol. 10, no. 43, pp. 20089–20095, Nov. 2018.

[133] M. Di Ventra, Y. V. Pershin, and L. O. Chua, "Circuit elements with memory: Memristors, memcapacitors, and meminductors," *Proc. IEEE*, vol. 97, no. 10, pp. 1717–1724, Oct. 2009.

[134] D. Jeltsema and A. van der Schaft, "Ideal memcapacitors and meminductors are overunity devices," *Sci. Rep.*, vol. 10, no. 1, pp. 1–8, Dec. 2020.

[135] S. H. Lee, X. Zhu, and W. D. Lu, "Nanoscale resistive switching devices for memory and computing applications," *Nano Res.*, vol. 13, no. 5, pp. 1228–1243, May 2020.

[136] W. Talsma, H. van Loo, S. Shao, S. Jung, S. Allard, U. Scherf, and M. A. Loi, "Synaptic plasticity in semiconducting single-walled carbon nanotubes transistors," *Adv. Intell. Syst.*, vol. 2, no. 12, 2020, Art. no. 2000154.

[137] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 712–727, Mar. 2006.

[138] D. E. Taylor, "Survey and taxonomy of packet classification techniques," *ACM Comput. Surv.*, vol. 37, no. 3, pp. 238–275, Sep. 2005.

[139] G. Sakellari, "The cognitive packet network: A survey," *Comput. J.*, vol. 53, no. 3, pp. 268–279, Mar. 2010.

[140] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Comput. Surv.*, vol. 38, no. 1, p. 1, Jun. 2006.

[141] W. Pan, Z. Li, Y. Zhang, and C. Weng, "The new hardware development trend and the challenges in data management and analysis," *Data Sci. Eng.*, vol. 3, no. 3, pp. 263–276, Sep. 2018.

[142] N. Bilton and J. Markoff, "A hardware renaissance in Silicon Valley," The New York Times, 2012, vol. 25. [Online]. Available: https://www.nytimes.com/2012/08/26/technology/silicon-valleys-hardware-renaissance.html

[143] H. Bauer, O. Burkacky, J. Kupferschmidt, and A. Rocha, "From hardware to software: How semiconductor companies can lead a successful transformation," *Semiconductors*, vol. 6, pp. 65–72, Jan. 2017.

[144] O. Krestinskaya, I. Dolzhikova, and A. P. James, "Hierarchical temporal memory using memristor networks: A survey," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 5, pp. 380–395, Oct. 2018.

[145] M. T. Arafin, C. Dunbar, G. Qu, N. McDonald, and L. Yan, "A survey on memristor modeling and security applications," in *Proc. 16th Int. Symp. Quality Electron. Design*, Mar. 2015, pp. 440–447.

[146] W. Woods, M. M. A. Taha, S. J. Dat Tran, J. Burger, and C. Teuscher, "Memristor panic—A survey of different device models in crossbar architectures," in *Proc. IEEE/ACM Int. Symp. Nanosc. Archit.*, Jul. 2015, pp. 106–111.

[147] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943.

[148] M. W. Roth, "Survey of neural network technology for automatic target recognition," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 28–43, Mar. 1990.
[149] J. J. Buckley and Y. Hayashi, "Fuzzy neural networks: A survey," *Fuzzy Sets Syst.*, vol. 66, no. 1, pp. 1–13, 1994.
[150] J.-Y. Potvin, "State-of-the-art survey—The traveling salesman problem: A neural network perspective," *ORSA J. Comput.*, vol. 5, no. 4, pp. 328–348, Nov. 1993.
[151] M. Stuart and M. Manic, "Survey of progress in deep neural networks for resource-constrained applications," in *Proc. 43rd Annu. Conf. IEEE Ind. Electron. Soc.*, Oct. 2017, pp. 7259–7266.
[152] Y. Goldberg, "A primer on neural network models for natural language processing," *Comput. Sci.*, vol. 57, pp. 345–420, Aug. 2016.
[153] A. Bhattacherjee, S. Roy, S. Paul, P. Roy, N. Kausar, and N. Dey, "Classification approach for breast cancer detection using back propagation neural network: A study," in *Deep Learning and Neural Networks: Concepts, Methodologies, Tools, and Applications*. Hershey, PA, USA: IGI Global, 2020, pp. 1410–1421.
[154] Y. S. Yang and Y. Kim, "Recent trend of neuromorphic computing hardware: Intel's neuromorphic system perspective," in *Proc. Int. SoC Design Conf. (ISOCC)*, Oct. 2020, pp. 218–219.
[155] W. Olin-Ammentorp and N. Cady, "Biologically-inspired neuromorphic computing," *Sci. Prog.*, vol. 102, no. 3, pp. 261–276, Sep. 2019.
[156] Y. Li, Z. Wang, R. Midya, Q. Xia, and J. J. Yang, "Review of memristor devices in neuromorphic computing: Materials sciences and device challenges," *J. Phys. D, Appl. Phys.*, vol. 51, no. 50, Dec. 2018, Art. no. 503002.
[157] R. A. Nawrocki, R. M. Voyles, and S. E. Shaheen, "A mini review of neuromorphic architectures and implementations," *IEEE Trans. Electron Devices*, vol. 63, no. 10, pp. 3819–3829, Oct. 2016.
[158] B. Rajendran and F. Alibart, "Neuromorphic computing based on emerging memory technologies," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 6, no. 2, pp. 198–211, Jun. 2016.
[159] R. Marani, G. Gelao, and A. G. Perri, "A review on memristor applications," 2015, *arXiv:1506.06899*.
[160] A. K. Maan, D. A. Jayadevi, and A. P. James, "A survey of memristive threshold logic circuits," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1734–1746, Aug. 2017.
[161] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, Nov. 2018, Art. no. e00938.
[162] A. P. Marugán, F. P. G. Márquez, J. M. P. Perez, and D. Ruiz-Hernández, "A survey of artificial neural network in wind energy systems," *Appl. Energy*, vol. 228, pp. 1822–1836, Oct. 2018.
[163] Z. Qin, F. Yu, C. Liu, and X. Chen, "How convolutional neural networks see the world—A survey of convolutional neural network visualization methods," *Math. Found. Comput.*, vol. 1, no. 2, pp. 149–180, 2018.
[164] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, "A survey of neuromorphic computing and neural networks in hardware," 2017, *arXiv:1705.06963*.
[165] C. D. James, J. B. Aimone, N. E. Miner, C. M. Vineyard, F. H. Rothganger, K. D. Carlson, S. A. Mulder, T. J. Draelos, A. Faust, M. J. Marinella, J. H. Naegle, and S. J. Plimpton, "A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications," *Biologically Inspired Cognit. Archit.*, vol. 19, pp. 49–64, Jan. 2017.
[166] C. Sung, H. Hwang, and I. K. Yoo, "Perspective: A review on memristive hardware for neuromorphic computation," *J. Appl. Phys.*, vol. 124, no. 15, Oct. 2018, Art. no. 151903.
[167] G. W. Burr, R. M. Shelby, A. Sebastian, S. Kim, S. Kim, S. Sidler, and K. Virwani, "Neuromorphic computing using non-volatile memory," *Adv. Phys., X*, vol. 2, no. 1, pp. 89–124, 2017.
[168] M. Davies, A. Wild, G. Orchard, Y. Sandamirskaya, G. A. F. Guerra, P. Joshi, P. Plank, and S. R. Risbud, "Advancing neuromorphic computing with Loihi: A survey of results and outlook," *Proc. IEEE*, vol. 109, no. 5, pp. 911–934, May 2021.
[169] V. Saxena, "Neuromorphic computing: From devices to integrated circuits," *J. Vac. Sci. Technol. B, Microelectron.*, vol. 39, no. 1, Jan. 2021, Art. no. 010801.
[170] G. Volanis, A. Antonopoulos, A. A. Hatzopoulos, and Y. Makris, "Toward silicon-based cognitive neuromorphic ICs—A survey," *IEEE Des. Test*, vol. 33, no. 3, pp. 91–102, Jun. 2016.

SAAD SALEH received the bachelor's and master's degrees in electrical engineering with majors in computer networks, communications, and electronics. Currently, he is doing research on cognitive and energy efficient in-network processing at the Groningen Cognitive Systems and Materials Center (CogniGron), University of Groningen, The Netherlands. His research interests include network systems, novel computing architectures, and software-defined networks.

BORIS KOLDEHOFE received the Ph.D. degree from the Chalmers University of Technology, Gothenborg. He is currently a Full Professor at the University of Groningen and an Adjunct Professor at the Technical University of Darmstadt. He worked formerly as a Postdoctoral Researcher and a Senior Researcher at the École polytechnique fédérale de Lausanne (EPFL) and University of Stuttgart. His research interests include networked and distributed systems, middleware, and event-based systems.