Memristor Based Neuromorphic Network Security System Capable of Online Incremental Learning and Anomaly Detection

Md. Shahanur Alam, Chris Yakopcic, Guru Subramanyam, and Tarek M. Taha
Dept. Of Electrical and Computer Engineering, *University of Dayton*
Dayton, OH, USA
{alamm8, cyakopcic1, gsubramanyam1, tarek.taha}@udayton.edu

***Abstract*—Real-time network intrusion and anomaly detection systems designed for battery powered devices are in high demand. This paper presents a study of unsupervised and supervised memristor based neuromorphic systems for such tasks. AutoEncoder (AE) and Multilayer Perceptron (MLP) algorithms are used to design memristor based intrusion and anomaly detection systems. The autoencoder shows strong intrusion detection performance with accuracy greater than 92.5% on zero-day attack packets. A real-time online incremental learning and anomaly detection system is also designed using the effective anomaly detection abilities of the AE. The learning system uses two autoencoders, one AE is pretrained for classifying network packets as normal and malicious, and the second AE is initialized with random weights and learns malicious data incrementally. Thus, this system is able to flag new attack classes during runtime. The real-time intrusion detection system performs with an accuracy greater than 89.7%. The memristor based implementation shows that the proposed system can be implemented using extreme low power for edge and IoT applications.**

Keywords—memristor, real-time, low power, anomaly detection

I. INTRODUCTION

Modern computing devices operating in the real world are exposed to the pervasive internet system, while continuously sharing a huge amount of data. The ubiquity of the internet also increases the possibility of theft of valuable and confidential information. According to [1], the minute a vulnerability is exposed to an intrusion, data confidentiality, integrity, and availability is at risk, which can cause an unrequitable loss for an organization. To protect a network from the threat of breaching, it needs a continuous surveillance system [2].

To increase internet security, there are two eminent intrusion detection systems available in the industry. One is a signature/rule-based system which is known as SNORT, and the other is an anomaly-based system [3]. SNORT compares an incoming packet with a list of stored signatures to detect if the packet is a threat to the network. The drawback of the system is that it cannot detect any new attacks, especially 'zero day' attacks [3]. The anomaly-based intrusion detection system compares the incoming packets with learned patterns to detect an intrusion. Neural network-based intrusion detection systems have shown promising results when performing anomaly detection. The neural networks exhibit adaptability and are able to learn in real-time [1], which makes these types of systems strong candidates when used for online learning and adaptable threat detection [4].

The learning mechanisms in neural networks are of two main types, supervised and unsupervised [5]. In supervised learning, the network uses labeled data for learning and classification. Some supervised learning algorithms include the multilayer perceptron (MLP) and the convolutional neural network (CNN) [6]. Unlike supervised systems, unsupervised learning does not require data labels to learn, and the systems are primarily used as generative models. Autoencoders (AEs) and generative adversarial networks (GANs) [6] are examples of unsupervised learning systems. Researchers have been investigating both supervised and unsupervised learning for intrusion detection applications, but supervised techniques are primarily good for detecting known attack categories. In the case of unsupervised learning, the network learns to find the anomalies within a set of packets, so this technique is more suitable for detecting categories that have not been predetermined [7]. The main bottleneck of a neural network system is the requirement of a large memory unit and high-power consumption during training, especially when graphics processing units (GPUs) are used for training the network [8].

Modern communication and computation have been shifting to battery powered smart devices like IoTs and edge computing modules. The IoTs are connected to the communication channel and continuously share a huge amount of data. These IoTs collect data on human health, living habits, and even location [9]. Real-time network monitoring and intrusion detection is essential for these low power systems capable of storing and transmitting personal data.

To enable neural network based online learning and real-time monitoring for battery powered devices, the memristor can be viable solution. Memristors are nano-scale non-volatile resistive memory devices which can retain learned information when removed from a power source [10]. Memristor based neuromorphic systems have been developed recently for different applications including network security [4,10,11], and image reconstruction and recognition [12,13]. These devices

978-0-7381-4307-1/20/$31.00 ©2020 IEEE

can be patterned with a very high density, and they are able to perform multiply- add operations with extreme efficiency in a highly parallel fashion [10-14].

In this work, we investigated both supervised (MLP) and unsupervised (autoencoder) neural network algorithms to determine their performance for network intrusion detection in two different use cases (using the NSL-KDD dataset). In the first case, we determined each network's ability to learn and recognize a known set of attacks. In the second case, each system was trained, and then evaluated with a dataset that contained some attacks types that were not learned previously, representing the case of zero day attacks. The algorithms developed were then ported to simulated memristor crossbar systems to show successful operation at extreme energy and area efficiency.

We also examined incremental online learning and real-time packet detection in this work. In the real-world, network packets are not labeled, so an unsupervised network is more suitable for online learning and new class identification. The proposed system uses a pretrained autoencoder, which is trained with only normal (benign) data packets. This AE is able to sort the normal and malicious packets. A second autoencoder learns from only the packets determined to be malicious in an incremental fashion, and this process fine tunes the network for anomaly detection in real-time.

This paper is organized as follows: Section II discusses related work, and Section III discusses the details of the NSL-KDD dataset. Section IV presents the memristor based intrusion detection circuits, and Section V presents the memristor based anomaly detection system. Intrusion and anomaly detection results are presented in Sections VI. Section VII presents the power, energy, and timing of the system and Section VIII provides a conclusion.

II. Related work

Researchers have been working on network intrusion detection systems since the inception of computer networking systems. Both unsupervised and supervised methods have been explored in software in the literature. Although, only a couple of papers have presented memristor based network intrusion detection systems. Work in [15] presents a ternary content addressable memory circuit to accelerate regular expression matching. A memristor based deep packet inspection (DPI) system is presented in [11] for high speed intrusion detection and classification. A low power, and high-density pattern matching system is presented in [16] for detection of network packets. Work in [10] presents a supervised MLP implemented using memristor crossbars and achieves more than 99% testing accuracy using the KDD Cup'99 dataset. A neuromorphic system is presented using a deep autoencoder for intrusion detection [8] and achieved greater than 90.12% accuracy when implemented using the TrueNorth neurosynaptic chip. In earlier work [4], we presented a memristor based unsupervised real-time intrusion detection system that achieved 92.91% accuracy using offline training. This model learns unknown packets in real-time and responds to the malicious packets only. Some works have also been published for supervised [17-29] and unsupervised learning [20,21] for intrusion detection software that is not tied to specific custom hardware.

Existing intrusion detection systems are primarily rule based, and thus not effective for the detection of 'zero day' attacks [3]. The capabilities of continual learning, fine-tuning, and transfer of knowledge are crucial computational learning techniques when a system is operating in the real world and processing a continuous stream of data [27]. Work in [28] presents a supervised online learning algorithm for MNIST data using Hedge Backpropagation. The autoencoder algorithm is used in real world applications for anomaly detection [29]. The extreme learning machine (ELM) algorithm is also proposed for real-time intrusion detection in [30]. Work in [2] describes a hierarchical temporal memory (HTM) based unsupervised real-time anomaly detection system proposed for monitoring a video stream.

In our proposed work, we studied MLPs and autoencoders for supervised and unsupervised learning respectively. Each of these designs were implemented using a simulated memristor crossbar, which allows us to study energy and timing metrics in this custom nanoscale low power hardware.

We perform intrusion detection based on a set of known attacks, but we also perform experiments where new attacks are introduced to the network during runtime. Furthermore, online training for intrusion detection is performed on a redefined model which is consistent with earlier work [4]. To the best of our knowledge, there is no other published work on the relative study of supervised and unsupervised intrusion detection in memristor based neuromorphic systems. The unsupervised online learning and anomaly detection methods for network security in neuromorphic systems we present are also novel.

III. NSL-KDD Dataset

NSL-KDD dataset contains samples of network data packets, and it is a revised version of the KDD Cup'99 dataset. This dataset has both training and testing portions and they consist of 125,973 and 22,544 samples respectively [22]. Both datasets are comprised of normal and malicious packets. Table I shows the number of normal and malicious packets in the training and testing datasets for each class. The number of attack types in the training dataset is 22, but there are 39 attack types in the testing dataset. This means the testing dataset has 17 more malicious datatypes when compared to the training dataset. The training set attack types are described in [4]. The attack types that are absent in the training data set but exist in the testing dataset include: *apache2*, *udpstorm*, *processtable*, *worm*, *mailbomb*, *mscan*, *saint*, *xlock*, *xsnoop*, *snmpguess*, *snmpgetattack*, *httptunnel*, *sendmail*, *named*, *ps*, *xterm*, and *sqlattack*. These new categories in the testing dataset will be a challenge for supervised learning systems, as they have not learned these new datatypes. However, an anomaly detection system may be more suited to the detection of these new attack types if they appear out of place when compared to normal network data.

Normal and malicious packets both have 43 attributes with nominal, binary, and numeric values [22]. The nominal attributes are at the 2nd position (protocol/type), the 3rd position (service), the 4th position (flag), and the 42nd position (attack type). The network packets need to undergo some preprocessing steps before they are fed into the network for training and testing. At first, the nominal attributes are replaced

with the integers. Then all features are compressed according to min-max normalization to bound each feature to a value within 0 to 1 (including the integer representations of features 2 through 4). The 42th position is represents the attack type and this feature is replaced with a 0 or a 1 for normal and malicious packets respectively. Example packets from the NSL-KDD dataset are shown in Figs. 1 (a) and (b), and the preprocessed version of these same packets are displayed in Figs. 1 (c) and (d). Table I displays the data breakdown of the two NSL-KDD data files (Test+ and Train+) used in this study.

```
0,tcp,http,SF,287,2251,0,0,0,0,0,1,0,0,0,0,0,0,0,0
,0,0,3,7,0.00,0.00,0.00,0.00,1.00,0.00,0.43,8,219,
1.00,0.00,0.12,0.03,0.00,0.00,0.00,0.00,normal,21
```
(a)

```
0,icmp,eco_i,SF,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,1,1,0.00,0.00,0.00,0.00,1.00,0.00,0.00,1,16,1.0
0,0.00,1.00,1.00,0.00,0.00,0.00,0.00,ipsweep,18
```
(b)

```
0,0,0.04347,0,2.17e-07,1.05e-05,0,0,0,0,0,1,0,0,0,
0,0,0,0,0,0,0,0.0156,0.01761,0,0.11,0,0,1,0,0.22,0.3
568,1,1,0,0.01,0.02,0,0,0,0,1
```
(c)

```
0,1,0.10,0,1.304e-08,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0.0019,0,0.00195,0,0,0,0,1,0,0,0.0039,0.0627,1,
0,1,1,0,0,0,0,0.857
```
(d)

Fig. 1. Example network packets (a) and (b) directly from the NSL-KDD dataset and (c) and (d) preprocessed normalized packets ready for training and evaluation.

TABLE I. NORMAL AND MALICIOUS PACKET DISTRIBUTION IN THE NSL-KDD TRAINING AND TESTING DATASETS

Class	Normal	DoS	Probe	R2L	U2R	Total
NSL-KDD Test+	9711	7460	2421	2885	67	22,544
NSL-KDD Train+	67343	45927	11656	995	52	125,973
NSL-KDD Test+ Variants	1	11	6	15	7	40
NSL-KDD Train+ Variants	1	6	4	8	4	23

IV. MEMRISTOR BASED INTRUSION DETECTION CIRCUITS

A. Network Topologies

The general architectures for the autoencoder and multilayer perceptron (MLP) are displayed in Fig. 2. An autoencoder is used as a generative model to recreate the trained samples after they are compressed at the bottleneck layer.

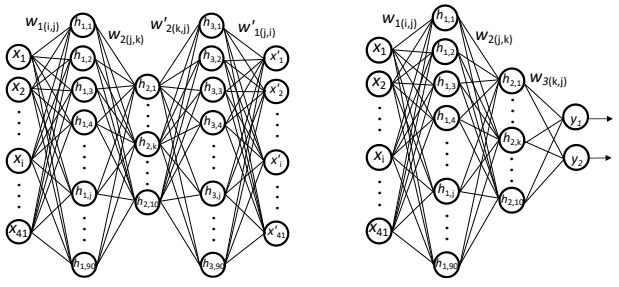

Fig. 2. Example architectures of (a) the autoencoder and (b) the MLP used in this study.

The mathematical details of the autoencoder are described in our earlier work [4]. The MLP is a supervised network that can classify the data samples within a known set of classes. Traversal of the MLP is carried out according to equations (1-3). Here b and $f(x)$ denote the bias and activation function respectively.

$$L_{1j} = f(\sum_{i=1}^{41} w_{1(i,j)} . x_i + b_{1j}) \quad (1)$$

$$L_{2k} = f(\sum_{j=1}^{90} w_{2(j,k)} . h_{1j} + b_{2k}) \quad (2)$$

$$L_{3j} = f(\sum_{k=1}^{10} w_{3(k,j)} . h_{2k} + b_{3j}) \quad (3)$$

B. Memristor Based Implementation

The memristor based autoencoder and MLP networks are based on previous works [10,12,13] for network intrusion and anomaly detection. Memristors [23] are a nano-scale non-volatile memory device that can be operated at extremely low power [12] when used to implement layers in neural networks. Thus, the memristor is a suitable candidate for use in embedded neuromorphic systems [13]. Memristors are essentially utilized to approximate the synaptic connectivity between neurons. Therefore, the memristor stores the connection strength between neurons and the incoming signals. Memristors laid out in a crossbar pattern can be used to store a weight matrix (or neural network layer) very effectively.

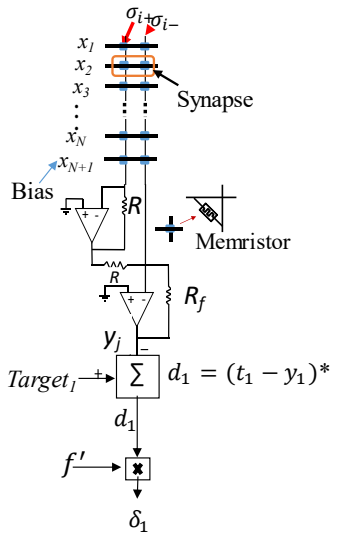

Fig. 3. Memristor based neuron circuit.

The basic neuron circuit we use in this work is displayed in Fig. 3, and this pattern can be repeated within a crossbar to define an entire layer of neurons. Thus, this memristor based neuron layer can perform multiply and addition operations very efficiently in parallel in the analog domain [24]. This neuron circuit requires two memristors to represent a single synaptic weight. For a given input, a positive weight value is observed when $\sigma_{i+} > \sigma_{i-}$, otherwise the weight observed is negative [12,13,24]. This circuit is able to carry out multiply-add operations according to equation (4). The memristor conductivity range we assumed for this work is as follows: $\sigma_{max} = 2 \times 10^{-5}\ \Omega^{-1}$ and $\sigma_{min} = 1 \times 10^{-7}\ \Omega^{-1}$.

$$DP_j = \sum_{i=1}^{N+1} x_i \times \left(\sigma_{ij}^{+} - \sigma_{ij}^{-}\right) \quad (4)$$

$$y_j = f(DP_j) \quad (5)$$

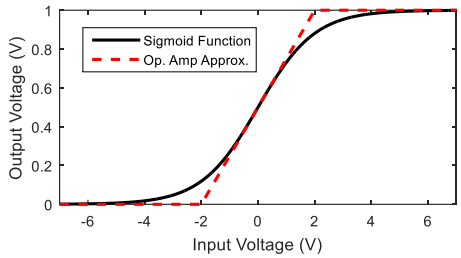

Fig. 4. Sigmoid activation function overlaid with the utilized approximation.

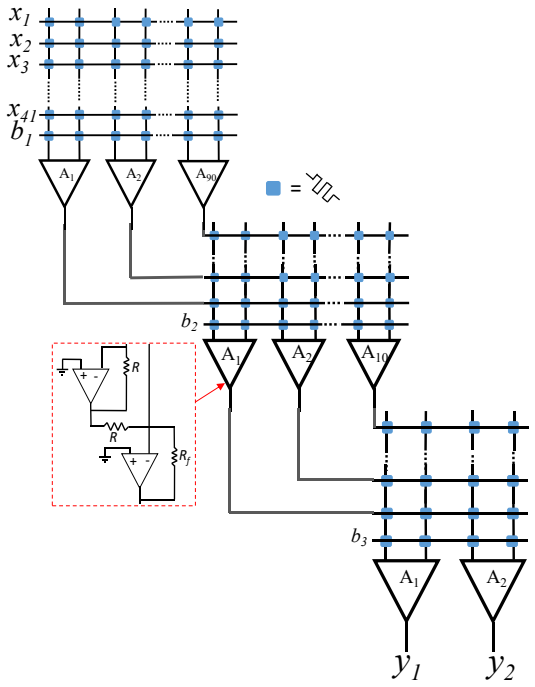

Fig. 5. Memristor crossbar circuits build a multilayered neural network with inputs (X) and outputs (Y). The Op-amp activation circuit is showing in the inset.

In Fig. 3, each input column is connected to a virtually grounded op-amp (operational amplifier). The output y_j in equation (5) represents the neuron output after accounting for the activation function, which is approximated using the rail voltages to effectively squeeze DP_j values. Equation (6) displays the typical sigmoid function used in deep learning. Equation (7) represents the approximated sigmoid activation function generated by op-amp circuits when the V_{DD} and V_{SS} of op-amps are connected to 0 and 1 respectively. The plot in Fig. 4 displays an overlay of $f(x)$ and $g(x)$ to provide a visual comparison of these two activation functions.

$$f(x) = \frac{1}{1+e^{-x}} \tag{6}$$

$$g(x) = \begin{cases} 1, & x > 2 \\ 0.25x + 0.5, & |x| \le 2 \\ 0, & x < 2 \end{cases} \tag{7}$$

The autoencoder and MLP networks are feed-forward networks. The multiple neurons interconnect and build a crossbar circuit, and the multilayered crossbar circuits construct a multilayer neural network circuit. Fig. 5 shows an MLP neural network with three crossbar circuits. In this study, we had to implement three crossbar circuits for the MLP and four for AE.

C. Memristor Crossbar Training

The architecture and training algorithm are similar for the autoencoder and the MLP. Although, the loss function calculation at the output layer of each is different. In the MLP, the output is compared with the true class label. However, in the autoencoder, the output vector is compared with the input vector. The MLP and autoencoder networks implemented in this experiment comprise two and three hidden layers, respectively. The training algorithm and training circuit have been adopted from [12,13,16,24]. Both the autoencoder and MLP are trained based on the Back Propagation (BP) algorithm [25]. The stochastic BP algorithm is utilized in these experiments where the synaptic weights are updated after each network packet is applied to the input layer instead of using a batch learning method. The training algorithm utilized is as follows.

i) Initialize the memristors with low random conductance.

ii) For an input network packet x:

a) Compute DP_j and y_j at the neuron outputs of each crossbar layer.

b) For a neuron j, compute the error δ_j based on the output y_j and the target t_j using equation (8) with the derivative of the activation function $g(x)$.

$$\delta_j = (x_j - y_j)g'(DP_j) \tag{8}$$

c) Back propagate the error toward the input from each hidden layer neuron j according to equation (9)

$$\delta_j = \sum_k \delta_k \, w_{k,j} g\left(DP_j\right) \tag{9}$$

d) Compute the weight update Δw_j according to the error function and update weight layers with learning rate η.

iii) Return to step (ii) until reaching the sufficient accuracy.

This simulation assumes on-chip learning is utilized, thus memristor resistance is tuned and optimized during the training process. On-chip training is beneficial because it accounts for the variation in resistance present across an array of memristor devices [26]. The memristor devices employed in this study have a resistance ratio of approximately 200 and a write threshold voltage of 1.3V.

V. Memristor Based Online Learning and Anomaly Detection Circuit

Our proposed system for online learning and real-time anomaly detection in network data is built on earlier work [4], but has been improved in its ability to perform effective online training. Fig. 6 shows how this system is able to complete online learning and anomaly detection. AE-1 is a pretrained autoencoder which sorts between normal and malicious packets.

Then, all malicious packets are sent to AE-2 which learns only malicious packets in real time. AE-2 is initialized with random weights and a random threshold. With every learning cycle, AE-2 fine tunes its knowledge and updates the threshold in real time. Fig. 7 shows the algorithm for online learning and

anomaly detection within the proposed system. Fig. 8 shows the circuit block diagrams of online Euclidean distance computing.

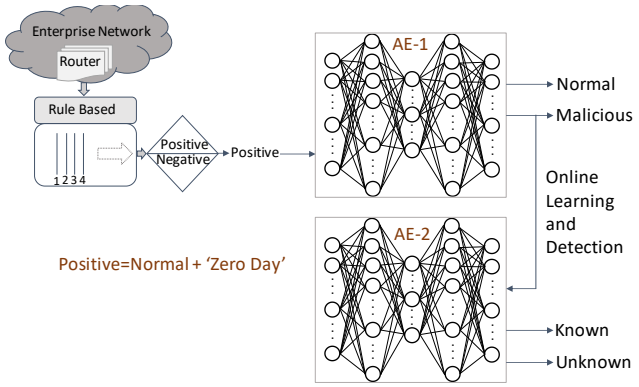

Fig. 6. The model for online learning and real-time anomaly detection.

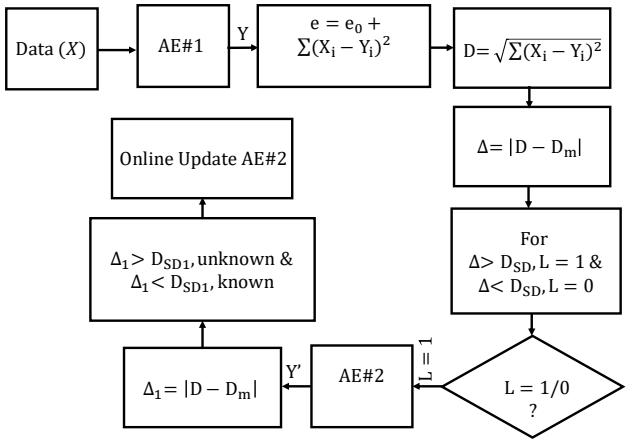

Fig. 7. Unsupervised, online learning, anomaly detection algorithm.

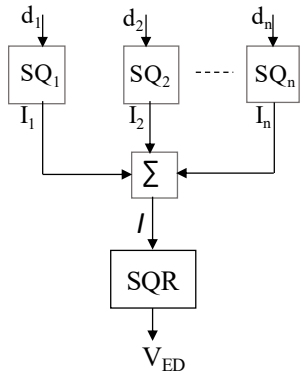

Fig. 8. Block diagram of the online Euclidean computing circuits for threshold optimization, *SQs* are the squaring circuits, and *SQR* is square root circuit. V_{ED} is the voltage equivalent of the Euclidean distance of two vectors.

From the classification result, the performance of the models will be evaluated according to the accuracy and sensitivity as in equations (12) and (13).

$$Accuracy = \frac{\sum True\ Normal + \sum True\ Malicious}{\sum Total\ Population} \quad (12)$$

$$Sensitivity = \frac{\sum True\ Normal}{\sum Predicted\ Normal} \quad (13)$$

VI. Results and Discussion

The presented work has two parts. First, we studied supervised and unsupervised techniques for network intrusion detection and observed performance in both traditional software and simulated memristor based systems. Second, we discussed anomaly detection results from the system described in Fig. 6.

A. Intrusion Detection Results

The NSL-KDD training dataset has 67,343 benign packets and 58,630 malicious packets of 22 different attack types. In our first study, the MLP is trained with 90% of the packets in the NSL-KDD Train+ data and evaluated with remaining 10% of the packets. The autoencoder is trained with 90% of the normal packets form the NSL-KDD Train+ dataset and evaluated with the remaining 10% of the normal packets, as well as 10% of the malicious packets from the Train+ data file. This study is performed to test each network's ability to classify known attack types.

A second study is then performed where the MLP network is trained with all 125,973 packets from the NSL-KDD Train+ dataset, and the autoencoder is trained with all 67,343 normal packets from this dataset. After which, each network is evaluated with the entire NSL-KDD Test+ dataset which contains 22,544 packets from 40 different datatypes (including normal data). This study was performed to determine each network's ability to classify zero day attacks, as the testing set contains 17 attack types not present in the training data.

The MLP examined has a layer structure of 41→90→10→2, and the autoencoder has a layer structure of 41→90→10→90→41. Each network is first implemented in traditional software for baseline testing. Then, each of these networks is implemented using the simulated memristor based hardware. Both execution types are implemented using MATLAB scripts. In the case of the hardware simulation, analog circuit equations are used to carry out the neural network traversal for realistic simulation.

Fig. 9 shows the mean squared error (MSE) during the training of the AE in both software and simulated memristor hardware when learning all normal packets in the NSL-KDD Train+ dataset. The memristor error curve is different most likely due to reduced dynamic range of memristor weight values and the approximated sigmoid function. The MSE of the MLP when learning the entire NSL-KDD training dataset is shown in Fig.10.

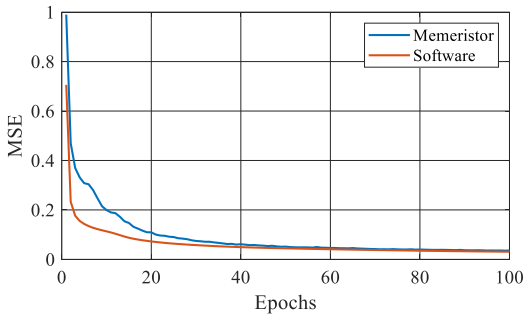

Fig. 9. Training error vs. epochs for the software and memristor based autoencoder design.

After training the networks, the autoencoder was evaluated according to the recognition datasets designed to test the autoencoder's ability to recognize both known and zero day attacks. Fig. 11 shows the binary classification results for normal and malicious network packets based on the threshold. The threshold parameter was computed and saved during the training session. Below the threshold margin in Fig. 11, all packets are classified as normal, otherwise they are considered malicious.

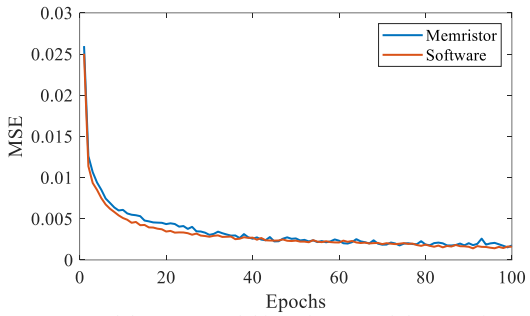

Fig. 10. MSE of the MLP model in software and the memristor system.

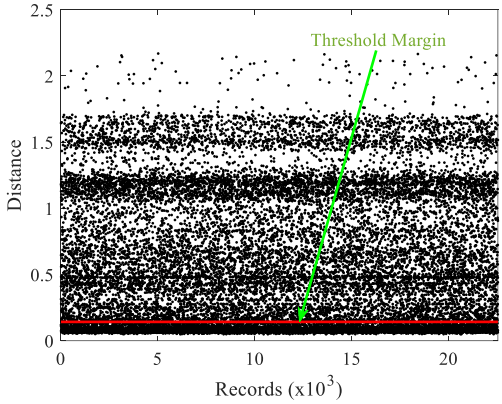

Fig. 11. The anomaly-based intrusion detection system in unsupervised autoencoder on NSL-KDD test dataset.

TABLE II. INTRUSION AND DETECTION RESULTS FOR THE AUTOENCODER SYSTEM.

Autoencoder	*Predicted Normal*	*Predicted Malicious*	*Accuracy*
Software	***Evaluated for Known Attacks***		
Actual Normal	TP=6309	FP=443	95.86%
Actual Malicious	FN=78	TN=5763	
Memristor	***Evaluated for Known Attacks***		
Actual Normal	TP=6125	FP=627	94.35%
Actual Malicious	FN=84	TN=5779	
Software	***Evaluated for Zero Day Attacks***		
Actual Normal	TP=8992	FP=719	*93.54%*
Actual Malicious	FN=737	TN=12096	
Memristor	***Evaluated for Zero Day Attacks***		
Actual Normal	TP=8732	FP=976	*92.52%*
Actual Malicious	FN=740	TN=12093	

The intrusion detection results are summarized in Tables II and III. In Table II the autoencoder shows strong performance when trained to recognize both known and zero day attacks. However, Table III shows that while the MLP is slightly better at recognizing known attacks compared to the autoencoder, the MLP is substantially worse at recognizing previously unlearned (or zero day) attacks.

The unknown attack types present in the NSL-KDD test dataset are unfamiliar to the MLP network. However, the autoencoder is trained only with normal packets to detect the anomalies, so this model successfully identified a high number of zero day attacks.

Fig. 12 summarizes the results obtained when comparing the experiments designed to recognize known attacks and zero day attacks in both software and the memristor system. These results again show that the MLP is effective when the testing environment is similar to its training environment but when the test samples are less familiar, the MLP performance deteriorates.

TABLE III. INTRUSION AND DETECTION RESULTS FOR THE MLP.

MLP	*Predicted Normal*	*Predicted Malicious*	*Accuracy*
Software	***Evaluated for Known Attacks***		
Actual Normal	TP=6740	FP=12	99.86%
Actual Malicious	FN=5	TN=5836	
Memristor	***Evaluated for Known Attacks***		
Actual Normal	TP=6736	FP=8	99.83%
Actual Malicious	FN=13	TN=5828	
Software	***Evaluated for Zero Day Attacks***		
Actual Normal	TP=9669	FP=42	82.4%
Actual Malicious	FN=3820	TN=9013	
Memristor	***Evaluated for Zero Day Attacks***		
Actual Normal	TP=9669	FP=48	79.4%
Actual Malicious	FN=4595	TN=8238	

**TN-True negative, TP-True Positive, FP-False Positive, FN-False Negative*

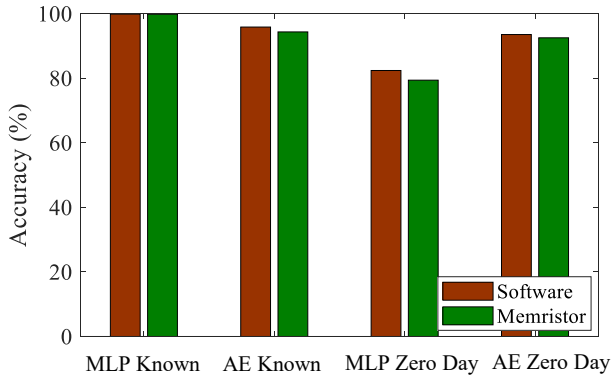

Fig. 12. Anomaly detection in supervised and unsupervised methods.

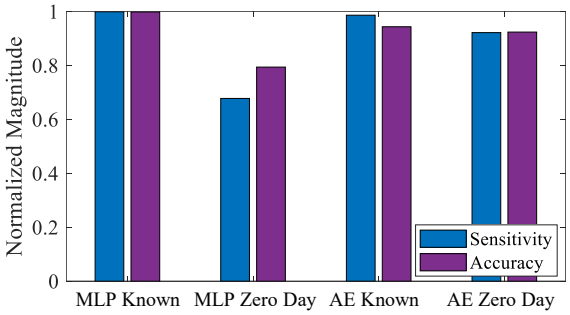

Fig.13. Sensitivity and Accuracy of MLP and AE in memristor for known and zero-day network packets

Fig. 13 shows the accuracy and sensitivity resulting from the memristor implementation of the MLP and AE networks. Unlike the MLP model, the autoencoder is trained to regenerate the normal packets, and it will likely regenerate malicious

packets poorly. Thus, the autoencoder performs classification of unknown attacks much better than the MLP when unknown (or zero day) attacks are passed through each system.

B. Anomaly Detection Results

The outperformance of autoencoder over the MLP for zero day attacks has motivated the design of the following real-time unsupervised learning and anomaly detection system. Fig. 13 displays the results of an experiment used to test this system. In the proposed design, the issue of catastrophic forgetting is reduced by sending all packets decided to be malicious by AE-1 to AE-2 for real-time training.

In an earlier design [4] AE-2 was initialized with a set of normal packets to set up the threshold and initialize the weights. Alternatively, in this model, AE-2 is initialized with random weights. As a result, the model did not recognize any packets during the initial cycle, and in successive cycles it continuously updated its network parameters as well as the threshold margin.

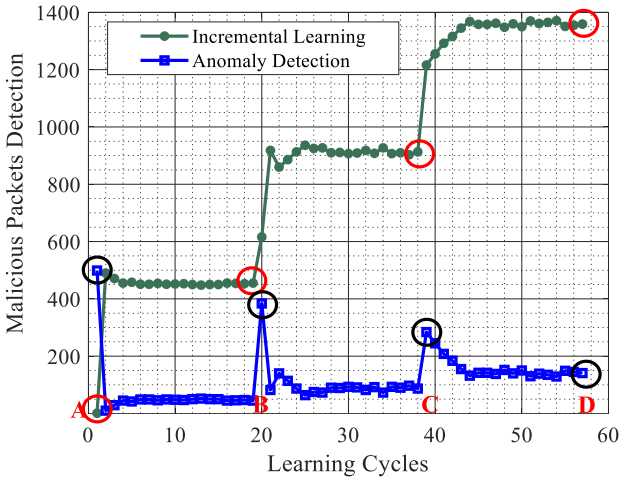

Fig. 14. Model of real-time learning and network intrusion detection system.

Fig. 14 shows the online learning and real-time detection result. The orange curve shows the result of anomaly detection on malicious packets over time, and the blue curve represents the incremental learning of malicious packets in real-time. At point **'A'** Test Set 1 is applied to the model, which contains 500 DoS type malicious packets as well as 500 normal packets. AE-2 learns only the malicious packets and considers the packets as learned if the network has already encountered the incoming malicious packet type. After the first cycle anomaly detection is reduced drastically, as the malicious packet type is no longer new to the anomaly detection system. At point **'B'** Test Set 2 is applied at the 19th cycle, and the number of anomalies detected increased to 390 from 45, as this new test set contains 500 attacks of a type that has never been presented to the system (in this case, the Probe attack type). The network fine tunes its parameters based on this new attack, and the detection threshold is also adjusted. At point **'C'** Test Set 3 is sent to the network and again anomaly detection increased at first, then reduced over time as the network got used to the new data type. The performance of the network is measured at points B, C, and D, and the results are presented in Table IV. Black circles on the anomaly detection curve indicate the emergence of new attacks. Red circles indicate the incremental learning progress on the learning curve. Fig. 15 shows the accuracy obtained for each of the three different test sets. The accuracy is computed at points B, C, and D after fine-tuning and rescaling the threshold of the system to observe the intrusion detection in real-time. The system exhibits 89.72% accuracy at the final point **'D.'**

TABLE IV. REAL-TIME LEARNING AND INTRUSION DETECTION RESULT

Test Set	Classes	Samples	FP	FN	TP	TN
Test Set 1	N+D	1000	10	45	490	455
Test Set 2	N+D+P	1500	52	87	448	913
Test Set 3	N+D+P+R	2000	65	141	435	1359

Note: N=Normal, D=DoS, P=Probe, R=R2L

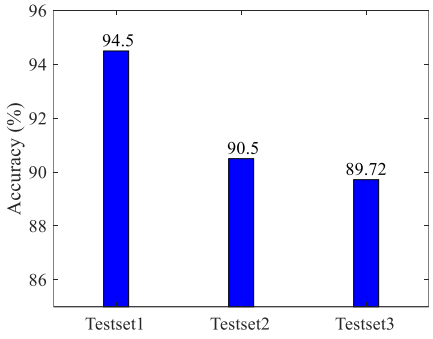

Fig. 15. Real-time anomaly detection accuracy for Test Sets 1 – 3.

VII. SYSTEM POWER, ENERGY AND TIMING ANALYSIS

The energy, power, and timing of the memristor crossbar and peripheral circuitry has been estimated for the pretrained autoencoder circuit for intrusion detection and for the anomaly detection system. The power, energy, and timing results are presented in Table V for each autoencoder in the anomaly detection circuit.

TABLE V. POWER, ENERGY, AND TIMING ANALYSIS OF THE PROPOSED NEUROMORPHIC SYSTEM

Metrics	AE Intrusion Detection Circuit	AE Anomaly Detection Circuit
Area (mm^2)	0.00135	0.00271
Training Power (mW)	20.6	28.6
Training Time (μs)	4.02	4.42
Training Energy: One Sample (nJ)	82	85
Recognition Power (mW)	7.56	15.1
Recognition Time (μs)	0.384	0.768
Recognition Energy: One Sample (nJ)	2.90	5.81

VIII. CONCLUSION

Unsupervised and supervised intrusion and anomaly detection systems have been studied in both traditional software and in simulated memristor neuromorphic hardware. The memristor crossbar technology exhibited the functionalities of the software-based model successfully. The experiment was performed in two different sub-experiments, one to examine learning of known attacks, and one to examine the learning of zero day attacks. The network intrusion detection is evaluated in the unsupervised autoencoder with an accuracy of 94.35% for the known attack experiment and 92.52% for the zero-day attack experiment. The supervised MLP performed

significantly worse when learning zero day attacks with an accuracy of 79.4%. The online incremental learning and anomaly detection system is implemented using a memristor hardware simulation and was evaluated in real-time with a final detection accuracy of 89.72%. In the future, we plan to study the supporting peripheral circuits for online learning, Euclidean distance and threshold computing, and backpropagation in more detail.

ACKNOWLEDGMENT

The work was supported through the National Science Foundation under grants 1718633.

REFERENCES

[1] Setareh Roshana, Yoan Michel, Anton Akusokd, Amaury Lendasse, "Adaptive and online network intrusion detection system using clustering and Extreme Learning Machines", Journal of the Franklin Institute Vol. 355, pp. 1752–1779, Iss. 4, March 2018J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Subutai Ahmada, Alexander Lavina, Scott Purdya, Zuha Agha, "Unsupervised real-time anomaly detection for streaming data", Vol. 262, pp. 134-147, Neurocomputing, 2017K. Elissa, "Title of paper if known," unpublished.

[3] Matilda Rhode b, Pete Burnap b, Kevin Jones, 'Early-stage malware prediction using recurrent neural networks', computers & security Vol. 77, August 2017, pp. 578-594

[4] MS Alam, B. Rasitha Fernando, Yassine Jaoudi, Chris Yakopcic, Raqibul Hasan, Tarek M. Taha, and Guru Subramanyam, Memristor Based Autoencoder for Unsupervised Real-Time Network Intrusion and Anomaly Detection, International Conference on Neuromorphic Computations (ICONS), July 2019, Knoxville

[5] Nathalie Japkowicz, 'Supervised Versus Unsupervised Binary-Learning by Feedforward Neural Networks', Machine Learning, Issue-42, pp. 97–122, Kluwer Academic Publishers,2001

[6] Md Zahangir Alom, Tarek M. Taha, 'Network intrusion detection for cyber security using unsupervised deep learning approaches', National Aerospace and Electronic Conference (NAECON), June 2017

[7] Zhaowei Qu, Lun Su, Xiaoru Wang, Shuqiang Zheng, Xiaomin Song, Xiaohui Song, 'A Unsupervised Learning Method of Anomaly Detection Using GRU', 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, China, 2018

[8] Md Zahangir Alom and Tarek M. Taha, "Network Intrusion Detection for Cyber Security on Neuromorphic Computing System", 2017 International Joint Conference on Neural Networks (IJCNN), 14-19 May 2017, Anchorage, AK, USA.

[9] Jacob Wurm1, Khoa Hoang1, Orlando Arias1, Ahmad-Reza Sadeghi2 and Yier Jin1, 'Security Analysis on Consumer and Industrial IoT Devices', 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), Macau, China, 2016

[10] Yakopcic and T. M. Taha, "Analysis and Design of Memristor Crossbar Based Neuromorphic Intrusion Detection Hardware," IEEE/INNS International Joint Conference on Neural Networks (IJCNN), pp. 1-7, Rio de Janeiro, Brazil, Julys, 2018

[11] Venkataramesh Bontupalli, Chris Yakopcic, Raqibul Hasan, and Tarek M. Taha. 2018. Efficient Memristor-Based Architecture for Intrusion Detection and High-Speed Packet Classification. *J. Emerg. Technol. Comput. Syst.* 14, 4, Article 41 (November 2018), 27 pages

[12] Raqibul Hasan, Tarek M. Taha, 'Memristor crossbar based unsupervised training', National Aerospace and Electronic Conference (NAECON), Dayton, USA, June 2015

[13] C. Yakopcic, M. Z. Alom, and T. M. Taha, "Memristor Crossbar Deep Network Implementation Based on a Convolutional Neural Network," IEEE/INNS International Joint Conference on Neural Networks (IJCNN), pp. 963-970, Vancouver, BC, July 2016.

[14] G. W. Burr, "Accelerating large-scale neural networks with analog non-volatile memory," *1st International Workshop on Memristive Devices for Neuronal Systems*, Kiel, Germany, September 2016.

[15] Catherine E. Graves , Can Li , Xia Sheng, Wen Ma, Sai Rahul Chalamalasetti, Darrin Miller , James S. Ignowski , Brent Buchanan, Le Zheng, Si-Ty Lam, Xuema Li, Lennie Kiyama, Martin Foltin, Matthew P. Hardy, and John Paul Strachan 'Memristor TCAMs Accelerate Regular Expression Matching for Network Intrusion Detection', IEEE transaction on nanotechnology, Vol. 18, pp. 963-970, 2019

[16] Raqibul Hasan and Tarek M. Taha, 'Memristor Crossbar Based Low Power Intrusion Detection Systems', 17th Int'l Conf. on Computer and Information Technology, Dhaka, Bangladesh, 22-23 December 2014

[17] Jiaqi Yan, Dong Jin, Cheol Won Lee and Ping Liu, 'A Comparative Study of Off-Line Deep Learning Based Network Intrusion Detection', 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), Prague, Czech Republic, August 16 2018

[18] Sheraz Naseer, Yasir Saleem, Shezad Khalid, Muhammad Khawar Bashir, Jihun Han, Muhammad Munwar Iqbal, and Kijun Han, 'Enhanced Network Anomaly Detection Based on Deep Neural Networks', IEEE access, Vol.6 Pages. 48231-48246, 2018

[19] R. Vinaykumer, Mamoun Alazab, K. P. Soman, Prabaharan Poornachandaran, Ameer Al-Nemrat,and Sitalaksmi Venkatraman, 'Deep Learning Approach for Intelligent Intrusion Detection System', IEEE Access, Vol.7, pages.41525-41550, 2019

[20] Fahimeh Farahnakian, Jukka Heikkonen, 'A Deep Auto-Encoder based Approach for Intrusion Detection System', International Conference on Advanced Communications Technology (ICACT), February 11-14, 2018, Chuncheon-si Gangwon-do, Korea (South)

[21] Zhaomin Chen, Chai Kiat Yeo, Bu Sung Lee, Chiew Tong Lau, 'Autoencoder-based Network Anomaly Detection', 2018 Wireless Telecommunications Symposium (WTS), 17-20 April 2018, Phoenix, AZ, USA

[22] NSL-KDD data: https://www.unb.ca/cic/datasets/nsl.html

[23] Leon O. Chua, 'Memristor-The Missing Circuit Element', IEEE Transactions on circuit theory, Vol.18, Issue.5, pp. 507-519, 1971

[24] Raqibul Hasan, and Tarek M. Taha, 'A Reconfigurable Low Power High Throughput Architecture for Deep Network Training', arXiv:1603.07400 [cs.LG]

[25] Russell, S. & Norvig, P. (2002). Artificial Intelligence: A Modern Approach (2nd Edition). Prentice Hall, ISBN-13: 978-01379039555.

[26] Raqibul Hasan and Tarek M. Taha, "Enabling Back Propagation Training of Memristor Crossbar Neuromorphic Processors", International Joint Conference on Neural Networks (*IJCNN*), Beijing, July 2014

[27] German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, Stefan Wermter, 'Continual lifelong learning with neural networks: A review', Elsevier, Neural Networks 113 (2019) 54–71

[28] Doyen Sahoo, Quang Pham, Jing Lu, Steven C. H. Hoi, 'Online Deep Learning: Learning Deep Neural Networks on the Fly', Twenty-Seventh International Joint Conference on Artificial Intelligence, Pages 2660-2666, Stockholm, 2018

[29] Poojan Oza and Vishal M. Patel, 'C2AE: Class Conditioned Auto-Encoder for Open-set Recognition', arXiv:1904.01198v1 [cs.CV] 2 Apr 2019

[30] Setareh Roshana, Yoan Michel, Anton Akusokd, Amaury Lendasse, "Adaptive and online network intrusion detection system using clustering and Extreme Learning Machines", Journal of the Franklin Institute Vol. 355, pp. 1752–1779, Iss. 4, March 2018