Position Papers of the 19th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 73–78 DOI: 10.15439/2024F8800
ISSN 2300-5963 ACSIS, Vol. 40

Resource efficient Internet-of-Things intrusion detection with spiking neural networks

Miloš Živadinović
0000-0002-0342-340X
Faculty of Organizational Sciences
University of Belgrade
Jove Ilića 154, 11000 Beograd, Republic of Serbia
Email: mzdv@protonmail.com

Dejan Simić
0000-0002-0744-5411
Faculty of Organizational Sciences
University of Belgrade
Jove Ilića 154, 11000 Beograd, Republic of Serbia
Email: dsimic@fon.bg.ac.rs

***Abstract*—Spiking neural networks are a novel implementation of artificial neural networks closely based on neurobiology. Our goal is to analyze and see the plausibility of spiking neural networks as intrusion detection models based on the BoT-IoT dataset under a limited set of circumstances. We created a spiking neural network classifier in PyTorch and snn-torch based on Leaky Integrate-and-Fire neurons that managed to get an F1 score of 0.957 on 10 000 samples of the BoT-IoT dataset and 240 hidden spiking neurons. We performed training on the CPU for 300 epochs and 10 simulation steps per epoch, utilizing Adam optimizer, cross-entropy loss and backpropagation as a learning algorithm. Lowering hidden spiking neuron count from 240 to 72 and sample size from 10 000 to 1 000, we were able to optimize training time by 84% and testing time by 57% while having an F1 score of 0.944. We present Loss, Receiver Operating Characteristics, and Precision-Recall curves for the two experiments and summarized data for additional experiments performed with different sample sizes and neuron counts. We conclude that spiking neural networks for intrusion detection represents a viable solution for training and classification on resource-constrained devices with limited samples. Further research steps are presented to improve performance.**

***Index Terms*—spiking neural networks, artificial intelligence, intrusion detection systems, internet of things**

I. Introduction

EVEN though a strict definition does not exist, Internet-of-things (IoT) can be defined as a set of connected devices with the goal of exchanging sensor and communication data between themselves to provide joint computing capabilities [1]. These devices often function in environments with constrained resources, such as electricity, processing speed, and memory. In 2023 there were 15.14 billion IoT devices active, with the trend increasing to more than 29 billion devices by the decade's end [2]. allowing potential malicious actors to launch attacks across different domains and infrastructure.

With the global number of IoT devices available, they present a formidable attack surface for targeted cyber attacks. The number of cyber attacks launched against IoT devices surpassed one hundred twelve million potential intrusions in 2022. [3], emphasizing a need for IoT device intrusion detection systems. Ideally, edge devices should handle intrusion detection for quick recognition and adequate accuracy.

Dorothy Denning defined the first known occurrence of intrusion detection systems (IDS) [4] in 1987. She defined intrusion detection systems as expert systems consisting of a tuple of six elements (Subjects, Objects, Audit records, Profiles, Anomaly records, and Activity rules with the role of detecting anomalies in computer system access. Expert system methodology was gradually improved and replaced by statistical analysis [5] and pattern-oriented intrusion detection [6], as well as machine learning and artificial intelligence methods [7]. Applying statistics, machine learning, and artificial intelligence allows greater autonomy for IDS systems and improves response times and detection rates with unknown intrusions.

Deploying IDS solutions to IoT devices has always been challenging due to the constraints of resources mentioned above, most notably available memory. Machine learning and statistics powered IDS solutions provide potential improvements for deployment on IoT devices, but the device's computing power often constrains them.

The concept of neuromorphic computing promises to improve upon these limitations. Neuromorphic computing is defined as the development of computer systems based on biological characteristics of neurons and nerve systems [8]. One of important neuromorphic computing concepts are spiking neural networks, which are more energy and efficient than their traditional counterparts, as shown in the Izhikevich model [9] which is used as the baseline for biological neuron simulation due to its biological plausibility.

Spiking neural networks can be traced back to the original discovery of neural spiking by Hodgkin and Huxley [10], which was later abstracted by Bonhoeffer [11] into a more general purpose model that handles different kinds of neuronal behavior. This model is known as the Bonhoeffer - van der Pol model, inspired by models of the human heart [12]. In 1982. Hopfield [13] defined the concept of a "Hopfield network," which represents the first artificial spiking neural network.

The key feature that provides efficiency is the concept of spiking (or action potentials) [10] which allow efficient signaling of changes between neurons. Each neuron has an activation function operating on the electrochemical interactions between the neuron and its environment. When the neuron reaches the activation potential threshold, it performs an electrical discharge to other connected neurons.

©2024, PTI

Topical area: Advanced Artificial Intelligence in Applications

Spiking neural networks on dedicated chips can utilize a maximum 65mW of power usage per 1 000 000 spiking neurons [14]. Asghar et al. [15] created a neuromorphic chip based on spiking neural networks that are even more resource efficient, consuming, on average, 1.06 mW of power. The Spiking-YOLO model has achieved similar resource efficiency [16] utilizing spiking neural networks for object detection.

Another key difference between spiking neural networks and traditional artificial neural networks is the existence of time as a component used for neuronal learning. Spike timing dependant plasticity (STDP) [17] is based on the duration before pre-synaptic and post-synaptic spikes and represents one of the key learning mechanisms in artificial and biological spiking neural networks. Time component utilization represents the side-effect spiking neural network models, which are based on systems of differential equations compared to other common artificial intelligence models. It is worth mentioning that, like modern artificial neural network learning, Lillicrap et al. [18] observed a variant of backpropagation as a learning mechanism for biological neurons.

With current knowledge, applying spiking neural networks to the domain of intrusion detection would allow us equal or better performance than current state-of-the-art solutions. The added benefit would be significantly less usage of power and computing resources, per current literature, making them suitable for IoT devices.

II. Current state of spiking neural networks and intrusion detection

The application of spiking neural networks in intrusion detection systems is a relatively new concept. One of the first usages dates back to 2014. spiking neural network concepts managed to get a 99.78% success rate when detecting failure rates in power systems [19].

Alom and Taha defined a way for autoncoders to be transformed into spiking neural networks to gain the benefits of neuromorphic computing [20]. Utilizing IBM TrueNorth [14], it was possible to have an intrusion detection system with 90.12% accuracy consuming less than 50 mW of power. This research also presents the method of converting traditional artificial neural networks to spiking neural networks, allowing potential performance improvements without retraining.

Zhou et al. introduced the first complete spiking neural network implementation for intrusion in 2020 [21]. The system in question consists of three layers with a total of 205 neurons, and it managed to reach 98.98% accuracy for intrusion detection.

Zarzoor et al. have applied spiking neural networks with decision trees for Internet-of-things attack classification [22] with 95% accuracy on attack classification from the IoT Botnet 2020 dataset.

Besides spiking neural networks, Hassini et al. [23] provide a solution based on deep learning for intrusion detection that reached 99.96% accuracy across 15 classes for edge IoT devices. Although unrelated to spiking neural networks, this solution represents one of the most advanced state-of-the-art approaches.

Encountered research so far focuses on spiking neural networks that were used for classification outside of Internet-of-things devices. Even though this approach proves their applicability, it does not factor in the possibility of IoT devices performing attack classification independently of a centralized classifier.

Table I contains summarized findings with F1 scores and accuracy, whichever metrics are available due to the quality of work.

TABLE I
Summarized findings from current state

Paper	Accuracy
Wang et al. [19]	99.78%
Alom and Taha [20]	90.12%
Zhou et al. [21]	98.98%
Zarzoor et al. [22]	95%
Hassini et al. [23]	99.96%

III. Experiment setup

We have used the BoT-IoT [24]–[27], [27], [28] dataset for our research due to its subject area of network traffic belonging to IoT devices. Due to resource constraints regarding computing power used for training, we have used the already prepared 5% subset of the BoT-IoT dataset to perform training and testing of the model. The dataset contains 3 668 522 entries, with 477 entries marked as non-malicious and the remaining marked as malicious, randomized and split into 80% of the dataset used for training and 20% of the dataset used for testing before further subsampling for our experiments. We randomized data before subsampling according to experiment requirements and applied mini-batching with a size of 100 samples per mini-batch as a way to optimize limited experiment resources.

We have not used spike trains for our experiments and have decided to use the numeric representation of data in order to simplify the experiment.

The BoT-IoT dataset contains 46 different features. In order to simplify handling, we have selected the subset of features shown in Table II:

TABLE II
Selected BoT-IoT features

Feature name	Feature description	Type
proto	Network traffic protocol	String
spkts	Source-to-destination packet count	Numeric
dpkts	Destination-to-source packet count	Numeric
srate	Source-to-destination packets per second	Numeric
drate	Destination-to-source packets per second	Numeric
state	Transaction state	String

We have performed one-hot encoding of the *proto* and *state* features for easier training, increasing the total input features to 18 presented in Table III. The final list of utilized features is presented below:

TABLE III
MODIFIED SELECTED BOT-IOT FEATURES

Feature name	Feature description	Type
arp	ARP protocol detected	Boolean
icmp	ICMP protocol detected	Boolean
tcp	TCP protocol detected	Boolean
udp	UDP protocol detected	Boolean
spkts	Source-to-destination packet count	Numeric
dpkts	Destination-to-source packet count	Numeric
srate	Source-to-destination packets per second	Numeric
drate	Destination-to-source packets per second	Numeric
acc	ACC state	Boolean
con	CON state	Boolean
eco	ECO state	Boolean
fin	FIN state	Boolean
int	INT state	Boolean
mas	MAS state	Boolean
req	REQ state	Boolean
rst	RST state	Boolean
tst	TST state	Boolean
urp	URP state	Boolean

The goal of our research was to classify attack subcategories according to the above features. There was a total of four attack subcategories identified in the BoT-IoT dataset:

- Denial of Service (DoS)
- Distributed Denial of Service (DDoS)
- Reconnaissance
- Theft

Due to the uneven distribution of attack subcategories, we have removed the Theft attack subcategory from the classification since its number of occurrences is much lower than that of other attack subcategories. Theft attack subcategory has the potential to skew results since we need more data to train for Theft subcategories (in the 5% dataset, the Theft subcategory appears in less than 0.01% due to rounding). We applied one-hot encoding to the remaining subcategories. Table IV shows the number of occurrences for attack subcategories:

TABLE IV
ATTACK SUBCATEGORY OCCURENCE

Attack subcategory	Number of occurrences	Percentage of dataset
DoS	1 650 260	44.99%
DDoS	1 926 624	52.52%
Reconnaissance	91 082	2.48%
Theft	79	Less than 0.01%

The classifier consists of two main Leaky Integrate-and-Fire layers (with decay rate $\beta = 0.95$ and 120 fully connected neurons per layer) joined with three linear transformation layers used for reshaping data. Although we have introduced the Izhikevich model previously in this paper, we have decided to use the Leaky Integrate-and-Fire spiking neuron model, which is easier to implement, albeit with less biological plausibility. This decision is because we do not require full biological plausibility for our use case, and the Leaky Integrate-and-Fire spiking neuron model is less complex than the Izhikevich model.

We applied backpropagation with temporal characteristics to update the weights after each epoch of the experiment. Biological spiking neurons operate with different learning methods, such as spike timing dependant plasticity (STDP) and its variants. This field is under constant research to determine how learning is performed between neurons; hence, there is no correct answer for using the learning method.

We applied surrogate gradients to improve backpropagation performance. In our case, we used the fast sigmoid surrogate gradient applied to the spiking neural network layers. We chose the Adam optimization algorithm [29] with $learning\_rate = 0.0001, \beta_1 = 0.9, \beta_2 = 0.999$ and cross-entropy loss [30] for calculating the value of loss.

The training lasted for 300 epochs, each epoch containing 10 discrete time steps as the temporal component. There were no optimizations regarding stop-loss values where training would halt. Training was performed by Intel i7-3630QM laptop CPU, without GPU usage. PyTorch version used was 2.1.0+cpu, and snntorch version used was 0.7.0.

IV. EXPERIMENT RESULTS

We executed the original experiment with 240 hidden neurons and 10 000 samples from the 5% BoT-IoT dataset with applied one-hot encoding transformations, resulting in an F1 score of 0.957. The loss curve is shown in Fig. 1. The noisiness shown inside loss curves is characteristic because the training was performed per epoch per set of discrete time steps. The loss gradient was calculated after each epoch, thus causing the noisiness inside every epoch.

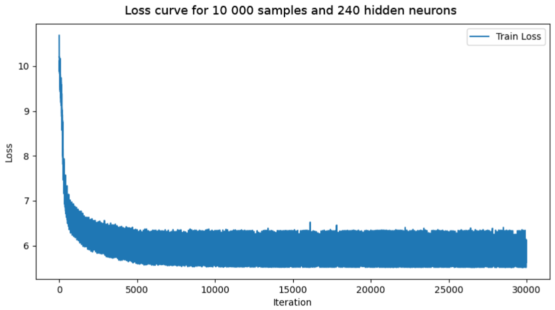

Fig. 1. Loss curve for 10 000 samples and 240 hidden neurons (two hidden layers of 120 spiking neurons)

ROC and PRC curves shown as Fig. 2. and Fig. 3 show additional experiment performance results. The area under Curve values is at the bottom for every attacking category.

Reconnaissance as the attack category was abbreviated to Recon due to brevity.

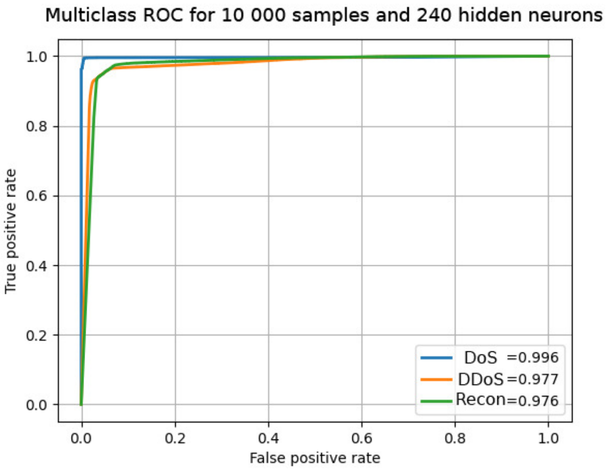

Fig. 2. Multiclass ROC curves for 10 000 samples and 240 hidden neurons (two hidden layers of 120 spiking neurons)

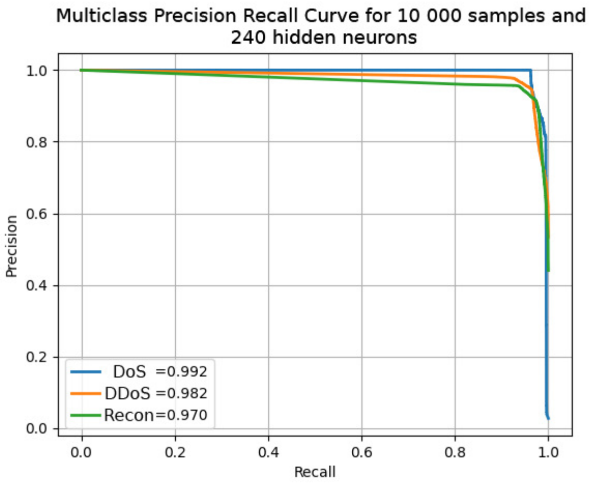

Fig. 3. Multiclass Precision-Recall curves for 10 000 samples and 240 hidden neurons (two hidden layers of 120 spiking neurons)

We have repeated the experiment with 1 000 samples and 72 spiking neurons to improve training and testing time. Our experiment yielded an F1 score value of 0.944.

The loss curve shown in Fig. 4 has a longer drop than the previous one with 10 000 samples. ROC (Fig. 5) and PRC (Fig. 6) curves show the impact of fewer samples and fewer neurons used for training, but showing similar results.

After promising results from experiments done with 1 000 and 10 000 samples and 72 and 240 hidden neurons, we have decided to start lowering the experiment variables, most notably sample size and hidden neuron count. Table V shows the results of previous experiments and additional ones performed with varying sample sizes and neuron counts.

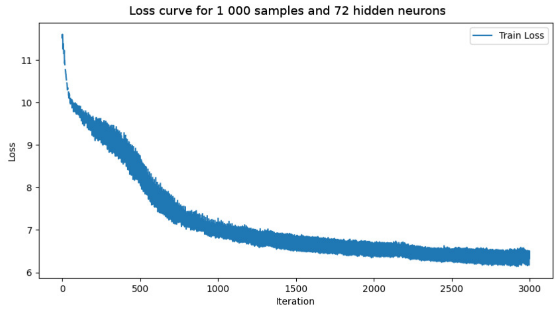

Fig. 4. Loss curve for 1 000 samples and 72 hidden neurons (two hidden layers of 36 spiking neurons)

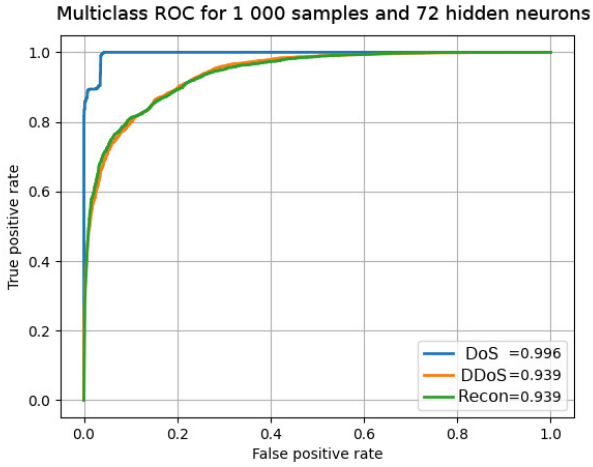

Fig. 5. Multiclass ROC curves for 1 000 samples and 72 hidden neurons (two hidden layers of 36 spiking neurons)

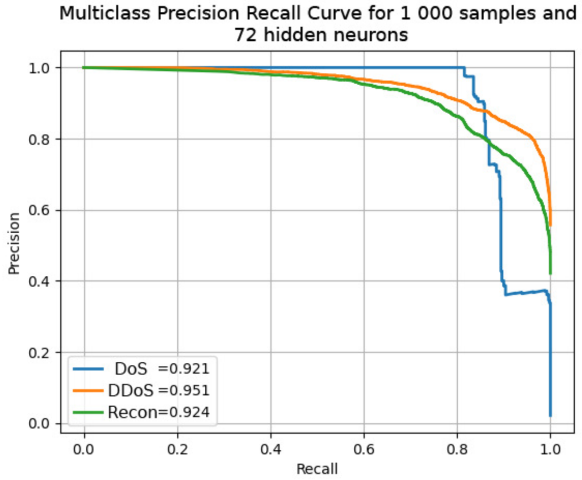

Fig. 6. Multiclass Precision-Recall curves for 1 000 samples and 72 hidden neurons (two hidden layers of 36 spiking neurons)

Sample sizes and neuron counts were randomly selected to see how the model would perform under new experimental conditions.

We can see that we get similar performance as state-of-the-art solutions with fewer samples and neurons that might apply to constrained systems. Previously described experiments with loss, ROC, and PRC charts are in bold as a baseline for other performed experiments that weren't presented in more detail.

TABLE V
SUMMARIZED EXTENDED EXPERIMENT DATA

Number of samples	Hidden neurons	Training time (seconds)	Testing time (seconds)	F1 score
500	20	58.730	0.091	0.910
500	240	103.546	0.149	0.949
1 000	20	145.789	0.085	0.915
1 000	**72**	**196.461**	**0.128**	**0.944**
1 000	240	231.894	0.159	0.959
10 000	72	1844.896	0.170	0.953
10 000	**240**	**2339.638**	**0.210**	**0.957**

Table VI below aggregates our experiment data with previously presented state-of-the-art data for comparison purposes.

TABLE VI
SUMMARIZED FINDINGS FROM CURRENT STATE

Paper	Accuracy
Wang et al. [19]	99.78%
Alom and Taha [20]	90.12%
Zhou et al. [21]	98.98%
Zarzoor et al. [22]	95%
Hassini et al. [23]	99.96%
1000 samples and 72 hidden neurons	95.99%
10 000 samples and 240 hidden neurons	96.33%

Following the experiment data and comparison with discovered state-of-the-art solutions, we can see that our two experimental models are comparative with state-of-the-art models, surpassing accuracy on some models, especially since our model is based on common hardware.

V. Discussion and future work

Experimental results from the previous section offer exciting insight into the applicability of spiking neural networks for intrusion detection systems and their behavior when the number of samples and neurons vary.

From our starting setup with 10 000 samples and 240 hidden neurons, we were able to lower the number of samples and neurons to 1000 samples and 72 neurons with similar F1 scores (0.957 versus 0.944, respectively). We performed additional experiments with even lower number of samples and hidden neurons that yielded F1 scores above 0.91 for all setups described in the tables above.

The first step should be to improve the computing resources used for experimentation and define stricter experimental criteria. Improvements would give us more detailed information on how the model performs with unconstrained sample sizes and diverse features and the inclusion of the Theft attack subcategory. Another task regarding computing resources would be accurate measurements of the spiking neural network to confirm resource efficiency, both on the original experiment environment and IoT devices. Experimental criteria should be stricter, and detailed experiment analysis should be performed on the behavior of false positives and false negatives, which can impact IoT devices differently due to their nature of being edge devices and with limited resources.

Besides resource increase, further research should be performed on different kinds of spiking neurons and different learning algorithms (such as STDP), as well as models with more than two hidden layers of spiking neurons.

Another interesting topic is the application of neuromorphic hardware such as Intel Loihi [31] or Graphcore's IPU processors [32] as integral components of the IoT device for intrusion detection and potential other artificial intelligence tasks that could be performed on-device. Utilizing neuromorphic hardware would improve total performance regarding accuracy and training times while allowing easier on-device intrusion detection.

VI. Conclusion

Internet-of-Things as a platform presents a new attack vector for malicious actors. Due to their decentralized and resource-constrained nature, performing adequate cyber attack detection and prevention without a centralized or significantly powerful device can be difficult.

We have introduced a method using spiking neural networks to enable intrusion detection classification in resource-constrained environments with cutting-edge performance. We performed testing using the BoT-IoT dataset, which includes a range of typical attacks found in Internet-of-Things environments and varying numbers of samples and hidden neurons. This allowed us to examine how F1 scores change as the number of samples and hidden neurons change, further optimizing performance with an F1 trade-off.

Even though we have reached state-of-the-art performance, we have presented additional steps in our research that can potentially improve performance by experimenting with different kinds of spiking neurons, more layers, and different learning algorithms.

References

[1] S. Li, L. D. Xu, and S. Zhao, "The internet of things: a survey," vol. 17, no. 2, pp. 243–259.

[2] L. S. Vailshery, "IoT connected devices worldwide 2022-2033."

[3] "Annual number of IoT attacks global 2022."

[4] D. Denning, "An intrusion-detection model," vol. SE-13, no. 2, pp. 222–232.

[5] T. Lunt, "Real-time intrusion detection," pp. 348–353. Conference Name: Digest of Papers. COMPCON Spring 89. Thirty-Fourth IEEE Computer Society International Conference: Intellectual Leverage ISBN: 9780818619090 Place: San Francisco, CA, USA Publisher: IEEE Comput. Soc. Press.

[6] S. Shieh and V. Gligor, "A pattern-oriented intrusion-detection model and its applications," pp. 327–342. Conference Name: 1991 IEEE Computer Society Symposium on Research in Security and Privacy ISBN: 9780818621680 Place: Oakland, CA, USA Publisher: IEEE Comput. Soc. Press.

[7] Wenke Lee, S. Stolfo, and K. Mok, "A data mining framework for building intrusion detection models," pp. 120–132. Conference Name: 1999 IEEE Symposium on Security and Privacy ISBN: 9780769501765 Place: Oakland, CA, USA Publisher: IEEE Comput. Soc.

[8] C. A. Mead, "Neuromorphic electronic systems," vol. 78, no. 10, pp. 1629–1636. MAG ID: 2163630896 S2ID: 459c554583c9a2f70dd36e84149989fde1e9f833.

[9] E. Izhikevich, "Simple model of spiking neurons," vol. 14, no. 6, pp. 1569–1572.

[10] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," vol. 117, no. 4, pp. 500–544. MAG ID: 1985940938.

[11] K. F. Bonhoeffer, "Activation of passive iron as a model for the excitation of nerve.," vol. 32, no. 1, pp. 69–91. MAG ID: 2019082810 S2ID: d4a34157d41b45efe5622fb11d29995ed0f26b82.

[12] B. van der Pol Jun Docts. Sc. and J. van der Mark, "LXXII. the heartbeat considered as a relaxation oscillation, and an electrical model of the heart," vol. 6, no. 38, pp. 763–775. MAG ID: 2071313546 S2ID: 9de0580210474428aee2312db59263647c568337.

[13] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," vol. 79, no. 8, pp. 2554–2558. MAG ID: 2128084896.

[14] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, G.-J. Nam, B. Taba, M. Beakes, B. Brezzo, J. B. Kuang, R. Manohar, W. P. Risk, B. Jackson, and D. S. Modha, "TrueNorth: Design and tool flow of a 65 mW 1 million neuron programmable neurosynaptic chip," vol. 34, no. 10, pp. 1537–1557. Conference Name: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

[15] M. S. Asghar, S. Arslan, and H. Kim, "A low-power spiking neural network chip based on a compact LIF neuron and binary exponential charge injector synapse circuits," vol. 21, no. 13, p. 4462.

[16] S. Kim, S. Park, B. Na, and S. Yoon, "Spiking-YOLO: Spiking neural network for energy-efficient object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 11270–11277. ISSN: 2374-3468, 2159-5399 Issue: 07 Journal Abbreviation: AAAI.

[17] S. Song, K. D. Miller, L. F. Abbott, and L. F. Abbott, "Competitive hebbian learning through spike-timing-dependent synaptic plasticity," vol. 3, no. 9, pp. 919–926. MAG ID: 1486852018.

[18] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, "Backpropagation and the brain," vol. 21, no. 6, pp. 335–346.

[19] T. Wang, G. Zhang, H. Rong, and M. J. Pérez-Jiménez, "Application of fuzzy reasoning spiking neural p systems to fault diagnosis," vol. 9, no. 6, p. 786.

[20] Z. Alom and T. M. Taha, "Network intrusion detection for cyber security using unsupervised deep learning approaches," pp. 63–69. MAG ID: 2790100928.

[21] S. Zhou, Shibo Zhou, Shibo Zhou, Xiaohua Li, Xiaohua Li, and X. Li, "Spiking neural networks with single-spike temporal-coded neurons for network intrusion detection.," ARXIV_ID: 2010.07803 MAG ID: 3093177876 S2ID: 70586ad226b9671b8c461f465850eff194eb726f.

[22] A. Zarzoor, N. Adnan, N. Al-Jamali, and D. Aldaloo, "Intrusion detection method for internet of things based on the spiking neural network and decision tree method," vol. 13, pp. 2278–2288.

[23] K. Hassini, S. Khalis, O. Habibi, M. Chemmakha, and M. Lazaar, "An end-to-end learning approach for enhancing intrusion detection in industrial-internet of things," vol. 294, p. 111785.

[24] N. Koroniotis, N. Moustafa, F. Schiliro, P. Gauravaram, and H. Janicke, "A holistic review of cybersecurity and reliability perspectives in smart airports," vol. 8, pp. 209802–209834. Conference Name: IEEE Access.

[25] N. Koroniotis and N. Moustafa, "Enhancing network forensics with particle swarm and deep learning: The particle deep framework."

[26] N. Koroniotis, N. Moustafa, and E. Sitnikova, "A new network forensic framework based on deep learning for internet of things networks: A particle deep framework," vol. 110, pp. 91–106.

[27] N. Koroniotis, N. Moustafa, E. Sitnikova, and J. Slay, "Towards developing network forensic mechanism for botnet activities in the IoT based on machine learning techniques," in *Mobile Networks and Management* (J. Hu, I. Khalil, Z. Tari, and S. Wen, eds.), Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 30–44, Springer International Publishing.

[28] N. Koroniotis, "Designing an effective network forensic framework for the investigation of botnets in the internet of things."

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization."

[30] D. R. Cox, "The regression analysis of binary sequences," vol. 20, no. 2, pp. 215–242. Publisher: [Royal Statistical Society, Wiley].

[31] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C.-K. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y.-H. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A neuromorphic manycore processor with on-chip learning," vol. 38, no. 1, pp. 82–99.

[32] G. , "IPU processors."