2024 17th IEEE/ACM International Workshop on Network on Chip Architectures (NoCArc) | 979-8-3315-0642-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/NoCArc64615.2024.10749907

NoCSNet: Network-on-Chip Security Assessment Under Thermal Attacks Using Deep Neural Network

Meisam Abdollahi*, Mohammad Chegini*, Mahdi Hasanzadeh Hesar†, Samaneh Javadinia*,
Ahmad Patooghy†, Amirali Baniasadi*
*Department of Electrical & Computer Engineering, University of Victoria, Victoria, Canada
†Department of Computer Systems Technology, North Carolina A&T State University, NC, USA
Email: {meisam, mchegini, samanehjavadinia, amiralib}@uvic.ca, mhasanzadehhesar@aggies.ncat.edu, apatooghy@ncat.edu

***Abstract*—As the demand for high-performance computing continues to rise, Network-on-Chip (NoC) architectures play a crucial role in enabling efficient data transmission within complex systems. However, the sensitivity of NoCs to intentional thermal fluctuations opens doors to conducting Denial of Service (DoS) attacks that can alter the system's reliability and security. In this paper, for the first time, we introduce NoCSNet as a novel database of NoC traffic collected under various network configurations and thermal attack scenarios. We also use Deep Neural Networks (DNNs) to analyze the collected traffic to enhance data transmission security in the presence of thermal DoS attacks. Through comprehensive experimentation and evaluation, we demonstrate the effectiveness of NoCSNet in capturing the security profile of NoC architectures, which can be actively used in protecting NoCs' data integrity and stability against thermal DoS attacks. The experimental results indicate that among the MLP, LSTM, and RNN deep neural networks, the RNN approach provides the highest attack detection accuracy of 93.8%. We anticipate that the collected dataset will help the community develop a deeper understanding of the susceptibility of NoCs against thermal DoS attacks.**

***Index Terms*—Thermal Attack, Deep Learning, Opto-electrical Network on Chip, Accuracy, Security Management**

I. Introduction

In modern Multi-Processor System-on-Chips (MPSoCs), inter-processor data exchange is mainly done through electronic Network-on-Chip (ENoC) that employ packet-based data transmission. This architectural approach offers notable enhancements over traditional bus and crossbar architectures by applying computer network theories and methodologies to on-chip communications [1]. However, recent advances in complementary metal oxide semiconductor (CMOS) circuits have prompted the integration of optical components into MPSoCs [2] i.e., optical NoCs (ONoCs) offering ultra-high bandwidth, low latency, and low power dissipation [2], [3]. As ONoC technology brings its challenges such as optical/electrical conversion overhead, complexities in laser integration, and sensitivity to thermal variations [4], [5], the hybrid opto-electrical paradigm appears to be an intriguing solution that offers advantages of both technologies [6].

ENoC and ONoC face an elevated risk of vulnerability to hardware attacks for several reasons. First, as a NoC interconnects various Intellectual Properties (IPs) including those produced in-house, sourced from trusted vendors, and IPs sourced from unverified vendors, there is always a possibility that a NoC can be leveraged for launching security attacks e.g., crypto-analysis attacks, side-channel attacks, denial of service (DoS) attacks, etc. [7], [8]. Such attacks can be conducted by Hardware Trojans (HTs) inserted in any of the sourced IP cores.

In this paper, we explore temperature variations as a security threat for targeting hybrid opto-electrical NoCs. We demonstrate how this vulnerability can be exploited by an adversary to compromise the network and exfiltrate sensitive information. Subsequently, we collect the necessary data in a simulated environment during the design phase to study the impact of thermal attacks. We utilize state-of-the-art neural networks in addition to traditional machine learning algorithms to analyze the behavior of such attacks. During runtime, we can predict whether each data transmission represents an attack or a safe transaction, enabling us to implement appropriate countermeasures. The contributions of this paper are as follows.

- In this paper, a novel security vulnerability affecting hybrid opto-electrical systems has been studied.
- We assess an opto-electrical NoC under various thermal attack scenarios along with normal data transmission during the design phase, using data collected from the Access-Noxim simulator.
- We utilize various neural networks and traditional machine learning approaches to identify the most accurate predictor in runtime scenarios.

The remainder of this paper is structured as follows. In Section II, the necessary background information on the network model and the thermal threat is provided. Section III explores the methodology including dataset gathering, pre-processing, and details of the neural network architecture. Section IV presents simulation experiments and analyzes the results, and finally Section V concludes the paper.

II. Targeted Architecture & Threat Model

In this section, we review the typical architecture for hybrid NoCs and then discuss the threat model proposed in [9] that is based on intentional thermal fluctuations occurring in the reviewed hybrid NoCs.

A. Opto-electrical Network-on-Chips

Typically, in a hybric NoC, long-distance communications (inter-cluster) occur through optical channels, while electrical communications are utilized for local destinations (intra-cluster) [6], [10]. Packet switching is used in the electrical domain, whereas optical circuit switching is utilized in the ONoC domain. When communication is necessary between two cores, the Network Interface (NI) sends a *path-setup* packet to the electrical router. Upon receiving this packet, the electrical router injects it into the electrical network and establishes an optical path for subsequent data transmission whenever an optical path is needed.

Optical signals are modulated at specific wavelengths, for instance, using Microring Resonators (MRs), which employ electrical signals for modulation. In this process, an array of photodetectors converts optical signals into corresponding electrical signals after they are multiplexed into a single optical waveguide [11]. This study adopts the cluster-based hybrid architecture, wherein optical routers are interconnected based on a mesh-based Crux optical router, with adaptations for a Wavelength Division Multiplexing (WDM) enabled network [9]. In MR technology, the resonant wavelength is highly susceptible to changes in the thermal conditions of the chip. To counteract the impacts of temperature fluctuations and variations induced during fabrication, MR necessitate tuning through the application of external current or heat (referred to as thermal tuning) to the MRs [12]. This adjustment process enables the modification of their effective refractive index. Within ONoCs, MRs undergo electrical and/or thermal tuning via a dedicated tuning circuit [2]. A similar circuit can be employed to activate and deactivate the MRs as required. However, these tuning circuits represent the most vulnerable aspect of the device concerning chip security.

Malicious hardware (such as a Hardware Trojan) could potentially manipulate the tuning circuit to interfere with the resonant wavelength of the ring resonators. Meanwhile, data traveling through the waveguide can be intercepted by malicious detector MRs without alteration, thereby posing a significant threat to photonic links.

B. Threat Model

The studied network comprises two layers: the initial layer constitutes an electrical network, while the upper layer is an optical network. Based on our assessments outlined in this section, an adversary can execute a thermal attack using any of the three scenarios described in [9], [13] to either drop or sniff data classified as private. The core of this attack lies in the thermal sensitivity of ONoCs. According to the proposed attack model, the attacker aims to overload a specific area of the network to raise the temperature of certain optical routers, thereby causing a shift in their operational wavelength. This deliberate congestion enables the attacker to manipulate the wavelength of the optical routers, granting access to data that the modified router would not otherwise have access to. The ensuing scenarios delve into how this thermal vulnerability can be exploited to compromise the security and dependability of ONoCs.

1) *NACK Replay Attack Scenario:* A malicious actor endeavors to identify the source of information and sends a series of NACK packets to the specified source router. This form of NACK replay attack capitalizes on the handshake process employed by the NoC in an end-to-end fashion. The replaying of NACK packets results in heightened network activity within the designated region, potentially elevating the local or overall chip temperature. Ultimately, this increased temperature may induce a wavelength alteration in the affected ONoC routers.

2) *Packet Drop Attack Scenario:* In instances where the attacker is unable to pinpoint the source address of a packet, they may opt to discard a segment of the packet, compelling the destination to issue a NACK to the source. This involves employing a fault injection attack to execute the ONoC thermal attack.

3) *False Traffic Injection Attack Scenario:* The attacker can inject significant volumes of fake traffic directly into the network to disrupt or overload a specific area, thereby breaching the thermal limits of the targeted ONoC router(s).

III. Data Collection and ML Processing

In a NoC-enabled MPSoC, following the generation of a message, the NoC routes the message based on the predefined routing algorithms, determining the path to the destination. This path typically involves multiple intermediary nodes, such as cores or routers, each tasked with forwarding the message progressively toward its final destination. In our method, as the message navigates through the network, it is subject to comprehensive monitoring where various attributes of the message and its transit are recorded. These attributes include the packet's size, source and destination identifiers, the chosen route, timestamps at various nodes, and traffic density, among others. Such monitoring is imperative as it furnishes the real-time data requisite for the analysis by a deployed learning algorithm (ML or DNN approach).

The overview of our proposed methodology is presented in Figure 1. We performed system-level simulations to assess the viability of the aforementioned attack scenarios. It should be noted that we did not alter the optical layer in any way; rather,

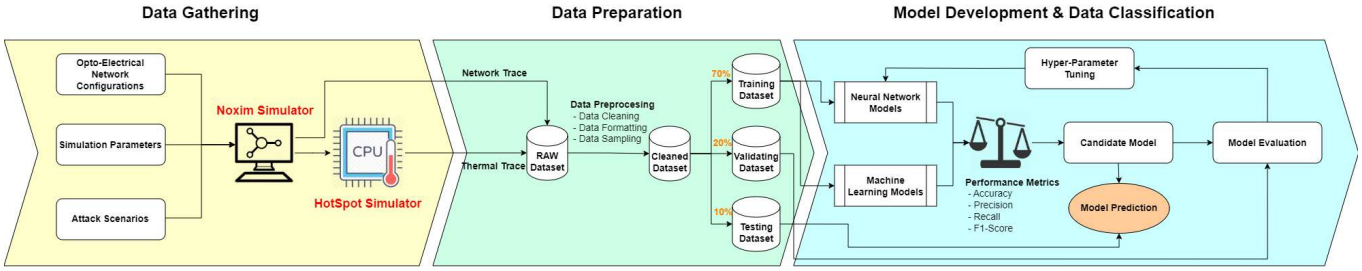

Fig. 1: Overview of the proposed methodology

TABLE I: Traffic features and value ranges applied in the model. *Source ID and Destination ID features are removed from the dataset for classification due to the applied secure countermeasure approach. **The attack label is the target that should be classified by the applied learning algorithm

Feature/Target	Value Range	Feature Description
Source ID*	0-35	The sender network node of a packet/flit
Destination ID*	0-35	The destination network node of a packet/flit
Current ID	0-35	The current network node where a packet/flit resides in each cycle
Flit Type	{head, tail, body}	Type of the flit
Hop Count	1-10	Number of hops from the current node to the destination
Flit Sequence Number	0-8	The unique sequence number for each flit of a packet
Packet Number	Depends on the Packet Injection Rate	The unique sequence number for each packet
Buffer Packet Count	0-112 (VC_Number (2) × Port_Number (7) × Buffer_Width(8))	Specifies the number of occupied buffers in the buffer of the current node
Input Port	{0: Local, 1: North, 2: South, 3: East, 4: West}	Port used by the flit to enter the current router
Output Port	(0: Local, 1: North, 2: South, 3: East, 4: West)	Port used by the flit to exit the current router
Core Temperature	From Ambient temperature to Maximum temperature	Temperature of the current node each packet/flit resides in each cycle
Current Cycle	1-1,500,000	The clock cycle of simulation
Attack Label**	{0,1}	Specifies whether this recorded data transmission is indicative of an attack or not

we solely conducted NACK replays in the electrical layer and monitored the resulting temperature changes in the optical layer. The findings indicate that the attack can significantly raise the network's temperature, leading to wavelength alterations in the optical routers. Throughout the system-level simulation of network parameters using the Noxim Simulator [14] and analysis of thermal behavior employing the Hotspot tool [15] which integrated into Access-Noxim simulator [16], we identified 13 essential features (as listed in Table I). Following data engineering and pre-processing of the dataset, various machine learning algorithms, as well as deep neural networks, were employed to identify the most effective classifier for distinguishing between attack and non-attack records based on the most significant metrics.

To collect the dataset, simulations were performed on a 6×6 mesh network featuring a secondary optical layer, spanning 1.5 million clock cycles with a 10% warm-up period. The electrical network parameters have been considered the default values in the Noxim simulator. The optical network parameters are imported into the simulator using the values reported in [10]. We recorded data while flits traversed routers by extracting information from electrical NoC packets. Approximately 40% of the traffic is related to attack transactions, which are randomly distributed from a random source; the remainder of the traffic is associated with normal uniform synthetic traffic. The dataset, comprising about 4.6 million records, is divided into three segments i.e., 70% allocated for training, 20% for validation, and 10% for testing. The details of the NoC traffic features used in our analysis are presented in Table I.

A. Data Preprocessing

As described in [13], in some state-of-the-art security approaches in on-chip data transmissions, the source and destination IDs are removed from the header of transmitted packets to prevent access by attackers, as we did in our training dataset. All 'not a number' (NaN) values and incomplete records were also removed from the dataset, and the final pruned dataset was normalized using the *StandardScaler* used in machine learning to standardize features by removing the mean and scaling them to unit variance. It transforms the dataset such that its distribution will have a mean value of zero and a standard deviation of one.

Figure 2 displays the correlation matrix of the dataset features. As depicted in the figure, the primary correlation exists between *Current Cycle* and *Packet Number*. This correlation arises from the observation that, as the simulation cycle progresses, the number of packets increases. Consequently, we have excluded the *Packet Number* feature from the feature list during the training of the ML algorithms. Another notable correlation is observed between *Temperature* and *Current Cycle*, which is expected since the advancement of the simulation typically leads to an increase in core temperature due to network activity. Since both features are important for

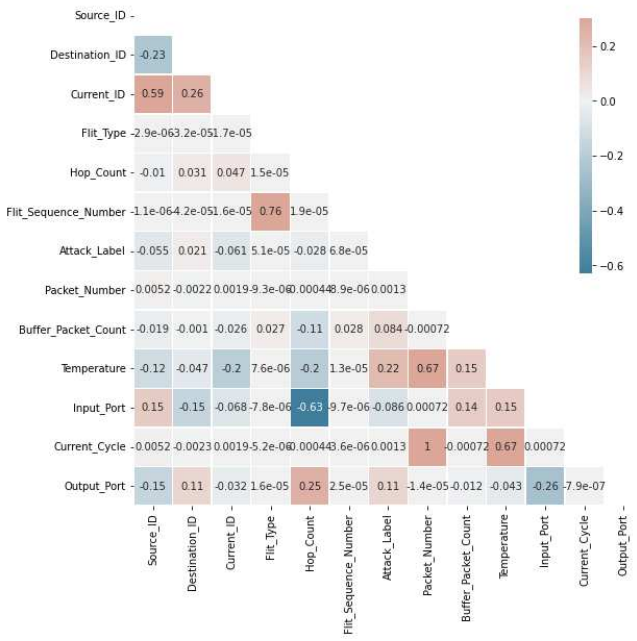

Fig. 2: Correlation matrix of features

our investigation, we will retain them both. Another notable correlation is evident between the *Source ID*/*Destination ID* and *Current ID* features. As previously mentioned, the *Source ID* and *Destination ID* features have been excluded from the dataset due to security concerns.

Table II illustrates the ranking of feature importance following the training of machine learning algorithms. The feature importance ranking provides valuable insights into the relevance and predictive power of different features across various machine learning algorithms. By examining the rankings assigned to each feature (i.e., a lower number indicates a higher importance rank), we can discern which features exert the most significant influence on the model's predictive performance. This analysis aids in understanding the underlying relationships within the dataset and helps identify key factors driving the outcomes of the model. Additionally, comparing feature importance across multiple algorithms offers a comprehensive perspective on feature relevance, highlighting consistent predictors across different modeling approaches. According to Table II, *Current ID* and *Input Port* emerge as two of the most critical features across all machine learning algorithms.

IV. Experimental Results

In this section, we compare the effectiveness of simple DNN, namely Multi-Layer Perceptron (MLP), RNN, and LSTM, with the most applicable machine learning classifiers such as XGBoost, LightGBM, Decision Tree, Random Forest, k-nearest neighbors (KNN), Naive Bayes, and Stochastic Gradient Descent (SGD) classifier. Figure 3 illustrates the variations in accuracy and loss functions during training and validation for various DNN approaches. It can be concluded that none of the algorithms are overfitted or underfitted based on the behavior of accuracy and loss functions. Additionally, the results indicate that, as previously mentioned, the data series approaches exhibit better parameters compared to the best MLP accuracy and loss values after 50 epochs.

A. Accuracy & Loss function Analysis

Table III depicts the most important performance metrics for comparing ML and DNN algorithms. The accuracy metric provides an overall measure of the model's correctness in predicting both attack and non-attack instances. A high accuracy score indicates that the model performs well in correctly classifying instances from both classes. Precision measures the proportion of true positive predictions among all positive predictions, highlighting the model's ability to avoid false positives. Similarly, recall, also known as sensitivity, quantifies the proportion of true positive instances correctly identified by the model among all actual positive instances, emphasizing the model's ability to detect attack scenarios. F1-score, the harmonic mean of Precision and Recall, provides a balanced measure of the model's performance, considering both false positives and false negatives. Analyzing these metrics collectively allows for a comprehensive assessment of the model's effectiveness in distinguishing between attack and non-attack scenarios. These metrics verify that although most ML algorithms have lower accuracy, precision, recall, and F1-score values compared to DNN algorithms, Decision Tree and Random Forest algorithms (with a maximum tree depth of 20) outperform in these metrics. With increasing the depth parameters to 25, the accuracy will be changed to over 99%. Further investigation and consideration are warranted regarding this issue, as the presence of excessively deep trees suggests the potential for overfitting the models to the training

TABLE II: Feature importance ranking for different classifiers

Features	ML Algorithm						
	XGBoost	LightGBM	Decision Tree	Random Forest	Naive Bayes	SGD	KNN
Flit Type	7	8	9	8	7	7	8
Hop Count	5	5	3	2	3	2	4
Flit Sequence Number	9	9	8	9	8	9	9
Current ID	1	1	1	1	1	1	1
Buffer Packet Count	6	6	7	7	4	4	5
Temperature	4	3	4	4	6	5	6
Input Port	2	2	2	3	2	3	2
Output Port	8	7	6	5	5	6	7
Current Cycle	3	4	5	6	9	8	3

TABLE III: Metrics comparison in ML and DNN algorithms

	Algorithm	Performance Metric			
		Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ML	XGBoost	90.8	87.6	91.6	89.6
	LightGBM	80.6	83.7	68.4	75.3
	Decision Tree	**92**	**88.5**	**93.8**	**91.1**
	Random Forest	91.2	88.1	91.1	90.9
	Naive Bayes	62.3	72.4	22.5	33.7
	SGD	65.3	63.8	46.7	53.6
	KNN	89.3	86.1	89.8	87.9
DNN	MLP	90.3	86.6	91.7	89.1
	LSTM	93.5	92.7	92.3	92.5
	RNN	**93.8**	**92.9**	**92.5**	**92.6**

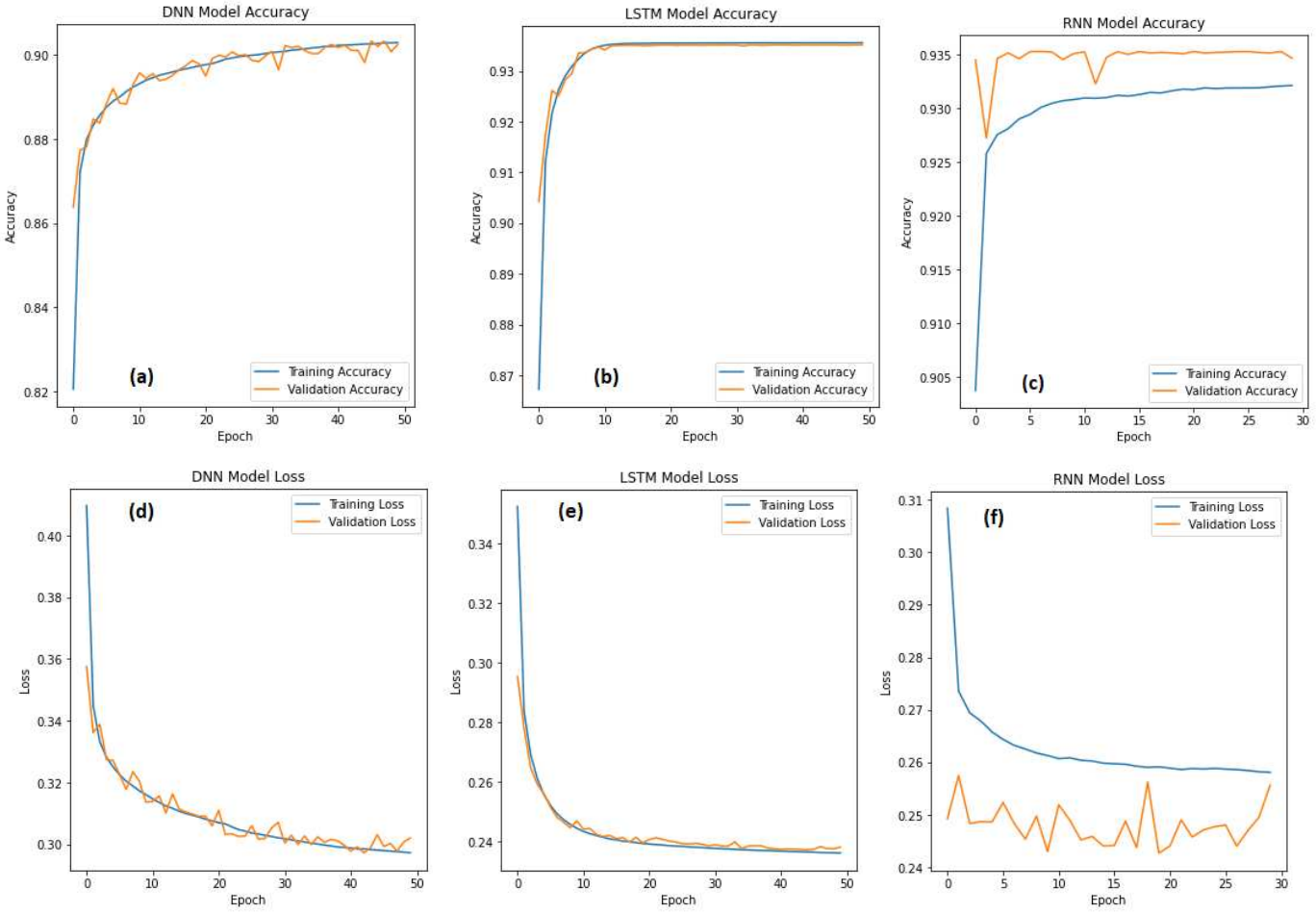

Fig. 3: Accuracy and loss parameters in each epoch of training and validation for different DNN approach (a) MLP Accuracy (b) LSTM Accuracy (c) RNN Accuracy (d) MLP Loss (e) LSTM Loss (f) RNN Loss

data. Overfitting occurs when the model learns the training data too well, capturing noise and outliers that are specific to the training set but do not generalize well to new, unseen data. To address this issue, it may be necessary to tune the hyperparameters of the decision tree and random forest algorithms, such as limiting the maximum depth of the trees or implementing pruning techniques to prevent overfitting and improve the generalization performance of the models.

On the other hand, each data record in the dataset was captured in every simulation cycle, and the dataset can be considered as a time series dataset (Current Cycle feature in Table I). The temporal dependencies and sequential nature of the data make RNNs and LSTMs well-suited for capturing patterns and making predictions. DNNs may struggle with such data because they do not consider the sequential information present in the data, whereas RNNs and LSTMs are designed to handle such dependencies effectively. Upon training three deep neural networks, it is imperative to compare their respective parameters, particularly the number of trainable parameters. This comparison sheds light on the complexity and capacity of each network to learn and represent the underlying patterns in the data. The MLP exhibits a simpler architecture, comprising densely connected layers, resulting in a relatively lower number of trainable parameters compared to LSTM and DNN. In contrast, LSTM, designed for sequential data analysis, incorporates specialized memory cells and gates, contributing to a larger parameter space. Meanwhile, DNN, characterized by its deep architecture with multiple hidden layers, tends to possess the highest number of trainable parameters among the three networks, facilitating the extraction of intricate features from the data. Regarding the number of hidden layers, the MLP typically has a single hidden layer, while the LSTM and RNN architectures can have multiple hidden layers. LSTM networks commonly include recurrent connections in multiple time steps, enabling them to capture long-term dependencies in sequential data effectively. In contrast, DNN architectures can be customized to include a variable number of hidden layers based on the complexity of the task and available resources. In terms of neurons per layer, MLP architectures often have a fixed number of neurons in each hidden layer, while LSTM

TABLE IV: The best models' parameters

Parameter	Neural Networks		
	MLP	LSTM	RNN
Learning Rate	0.01	0.001	0.001
Training Time (minutes)	640	870	910
Number of Layers	7	8	8
Number of Training Parameters	1,156,291	1,232,750	1,289,228

and DNN architectures can vary significantly depending on the specific architecture and task requirements. LSTM networks have a larger number of neurons per layer compared to MLPs due to the additional complexity introduced by memory cells and gates.

The experimental results of various machine learning algorithms alongside three DNN architectures, i.e. MLP, LSTM, and RNN, at design time for thermal attack detection in NoC systems, have yielded insightful results. Among these methods, RNNs and specifically LSTM architectures have demonstrated superior performance across multiple performance metrics. These findings underscore the suitability of RNN-based approaches, particularly LSTM, as robust security measures for real-time application in NoC systems. By leveraging the inherent capability of LSTM networks to capture temporal dependencies and patterns in sequential data, such approaches exhibit enhanced sensitivity to detect anomalous thermal behavior indicative of potential attacks. Furthermore, the effectiveness of these methods in accurately identifying and mitigating security threats at runtime highlights their potential for proactive defense mechanisms in NoC environments, thus enhancing the resilience and security posture of these critical systems.

B. Hyperparameter Tuning

To optimize a model, it is necessary to fine-tune its hyperparameters to minimize testing errors. Each classifier uses a unique number and variety of these parameters. Having experimenting with various permutations and combinations of optimal parameters, including the number of hidden layers, neurons per layer, dropout rates, batch size, learning rate, activation functions, and loss functions, we have successfully optimized neural network models to minimize the loss metric.

Table IV illustrates the key parameters of the optimal neural network models. In particular, the number of hidden layers ranged from 1 to 20. The dropout rate varied from 0.1 to 0.8 in increments of 0.1, and the number of neurons per hidden layer was set between 1 and 100. Additionally, the activation functions for each layer included Linear, Tanh, Relu, and Sigmoid. The learning rates were chosen from 0.1, 0.01, and 0.001. The classification loss function was Binary Cross-entropy, and the Adam optimizer was employed.

V. Conclusions

The susceptibility of NoCs to thermal Denial-of-Service (DoS) attacks presents a formidable challenge to system reliability and security. Through the development and evaluation of a NoC traffic dataset (the so-called NoCSNet), we presented a pioneering approach for analyzing NoC traffic using Deep Neural Networks (DNNs). We have demonstrated the efficacy of our framework in fortifying data transmission security in the face of thermal-based attacks. In our future work, we plan to deploy HDL design of NoCSNet to develop hardware-accelerated run-time component of attack detection and also an appropriate countermeasure.

References

[1] S. Kundu and S. Chattopadhyay, *Network-on-chip: the next generation of system-on-chip integration*. Taylor & Francis, 2014.

[2] K. Bergman, L. P. Carloni, A. Biberman, J. Chan, and G. Hendry, "Photonic network-on-chip design," 2014.

[3] M. Balti and A. Jemai, "Performance survey of classic and optic network-on-chip," *IET Circuits, Devices & Systems*, vol. 15, no. 4, pp. 393–402, 2021.

[4] A. Karabchevsky, A. Katiyi, A. S. Ang, and A. Hazan, "On-chip nanophotonics and future challenges," *Nanophotonics*, vol. 9, no. 12, pp. 3733–3753, 2020.

[5] M. Baharloo, M. Abdollahi, and A. Baniasadi, "System-level reliability assessment of optical network on chip," *Microprocessors and Microsystems*, vol. 99, p. 104843, 2023.

[6] M. Abdollahi, A. Namazi, and S. Mohammadi, "Clustering effects on the design of opto-electrical network-on-chip," in *2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, 2016, pp. 427–430.

[7] S. Charles and P. Mishra, "A survey of network-on-chip security attacks and countermeasures," *ACM Computing Surveys (CSUR)*, vol. 54, no. 5, pp. 1–36, 2021.

[8] P. Mishra and S. Charles, *Network-on-chip security and privacy*. Springer, 2021.

[9] M. Hasanzadeh, M. Abdollahi, A. Baniasadi, and A. Patooghy, "Thermo-attack resiliency: Addressing a new vulnerability in opto-electrical network-on-chips," in *25th International Symposium on Quality Electronic Design (ISQED'24)*. IEEE, 2024, pp. 1–6.

[10] M. Abdollahi, Y. Firouzabadi, F. Dehghani, and S. Mohammadi, "Thamon: Thermal-aware high-performance application mapping onto opto-electrical network-on-chip," *Journal of Systems Architecture*, vol. 121, p. 102315, 2021.

[11] F. Dehghani, S. Mohammadi, B. Barekatain, and M. Abdollahi, "Ices: an innovative crosstalk-efficient 2× 2 photonic-crystal switch," *Optical and Quantum Electronics*, vol. 53, pp. 1–15, 2021.

[12] Y. Ye, J. Xu, X. Wu, W. Zhang, X. Wang, M. Nikdast, Z. Wang, and W. Liu, "System-level modeling and analysis of thermal effects in optical networks-on-chip," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 2, pp. 292–305, 2012.

[13] A. Patooghy, M. Hasanzadeh, A. Sarihi, M. Abdelrehim, and A.-H. A. Badawy, "Securing network-on-chips against fault-injection and crypto-analysis attacks via stochastic anonymous routing," *ACM Journal on Emerging Tech. in Computing Systems*, vol. 19, no. 3, pp. 1–21, 2023.

[14] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, "Noxim: An open, extensible and cycle-accurate network on chip simulator," in *2015 IEEE 26th international conference on application-specific systems, architectures and processors*. IEEE, 2015, pp. 162–163.

[15] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "Hotspot: A compact thermal modeling methodology for early-stage vlsi design," *IEEE Transactions on very large scale integration (VLSI) systems*, vol. 14, no. 5, pp. 501–513, 2006.

[16] K.-Y. Jheng, C.-H. Chao, H.-Y. Wang, and A.-Y. Wu, "Traffic-thermal mutual-coupling co-simulation platform for three-dimensional network-on-chip," in *Proceedings of 2010 international symposium on VLSI design, automation and test*. IEEE, 2010, pp. 135–138.