Zero-Memory-Overhead Clipping-Based Fault Tolerance for LSTM Deep Neural Networks

Bahram Parchekani[1], Samira Nazari[1], Mohammad Hasan Ahmadilivani[2], Ali Azarpeyvand[1,2], Jaan Raik[2], Tara Ghasempouri[2], and Masoud Daneshtalab[2,3]

[1]University of Zanjan, Zanjan, Iran
[2]Tallinn University of Technology, Tallinn, Estonia
[3]Mälardalen University, Västerås, Sweden
[1]{bahram.parchekani, samira.nazari, azarpeyvand}@znu.ac.ir
[2]{mohammad.ahmadilivani, ali.azarpeyvand, jaan.raik, tara.ghasempouri}@taltech.ee
[3]masoud.daneshtalab@mdu.se

***Abstract*—Long Short-Term Memory (LSTM) Deep Neural Networks (DNNs) have shown superior accuracy in predicting and classifying time-series data. This has made them suitable for many applications, including safety-critical ones, such as healthcare, where fault tolerance is a major concern. Until now, fault resilience and mitigation in LSTMs are not thoroughly explored, raising concerns about exploiting them in safety-critical use cases. This work, first, extensively explores the effect of faults on LSTM DNNs using fault injection into parameters. Moreover, the paper presents two effective zero-memory-overhead fault tolerance techniques for LSTM DNNs to protect them against random faults in their parameters. Experimental results indicate that the proposed techniques can improve fault tolerance of LSTM-based DNNs up to 278.6 times with respect to unprotected ones.**

***Index Terms*—Hardware Reliability, Fault Tolerance, Neural Networks, LSTMs, Healthcare.**

I. Introduction

Deep Learning (DL) is being increasingly employed in safety-critical applications, e.g., healthcare and automotive, due to their outstanding accuracy in classification, prediction, etc. [1]–[3]. In this domain of applications, hardware reliability is a significant concern [4]–[6]. Hardware reliability is defined as the probability of hardware performing correctly with the presence of faults. Faults may occur due to temperature variation, aging, soft errors, etc., and flip the bits in logic or memory [4], [6]. Such an effect may lead to catastrophic results in safety-critical applications. With transistor scaling, fault rates in memories have been increased, which threatens the hardware reliability significantly [7], [8].

Healthcare applications exploit Deep Neural Networks (DNNs) extensively for various tasks such as diagnosis, treatment, and prediction of diseases and anomalies [9], [10] because of their outstanding strength in processing time-series data [11], [12]. Long Short-Term Memory (LSTM) neural networks are a subset of Recursive Neural Networks (RNNs) that are shown to be remarkably effective in classifying and predicting time-series data. They retain long-term information through time via feedback loops making them highly desirable for disease prediction in healthcare applications [12].

Throughout the literature, several research works have thoroughly studied the reliability of DNNs in safety-critical applications [5], [6]. Fault injection at the software simulation level is the predominant method for analyzing and evaluating the reliability of DNNs due to their fast execution time [4]. The related studies indicate that faults in memory impacting the parameters of DNNs result in a substantial reduction of their accuracy [8], [13].

Therefore, various methods for improving their fault tolerance are proposed. Fault tolerance techniques are applied either at the CNN's hardware level [14]–[16] or model level [17]–[19] and greatly impact their resilience against faults. Hardware-level techniques implement hardware redundancy in the accelerators, which require hardware redesign. Whereas in several scenarios it is impossible to modify the hardware, such as off-the-shelf processors and IPs. Hence, model-level techniques are applicable, which implicitly improve the fault tolerance of DNNs by introducing additional functionality within them.

Nearly all existing works study the reliability of Convolutional Neural Networks (CNNs) for image classification and object detection tasks [4]. Although LSTMs are widely deployed in safety-critical applications, including healthcare, their reliability against faults and fault tolerance methods are not extensively explored [4].

In a recent research work, the effect of faults in the computational units of LSTMs for image classification in automotive is studied and a hardware-level fault tolerance approach is proposed [20]. This method is shown to be effective, yet with up to 54% area overhead to the LSTM neural networks. In another prior work, a fine-grain and comprehensive resilience analysis for different sets of parameters in LSTM neural networks is performed [21]. It is indicated that the recurrent parameters in LSTMs are the most vulnerable components to faults since they affect the memory characteristics of LSTMs. Moreover, it is shown that the resilience can be remarkably improved by detecting the faulty weights and setting them to 0.

Regarding the literature, fault tolerance approaches for LSTMs are not extensively explored. The shortcomings in the previous works are:

- While the structure and applications of LSTM-based DNNs are various, only the resilience of small and simple LSTMs is studied for a limited number of applications. As in CNNs, resilience studies of Deep LSTM-based neural networks are required to be carried out to thoroughly analyze their reliability,
- Various fault tolerance techniques are required to be explored for LSTMs to identify the most efficient approaches for them.
- Reliability analysis metrics for image classification and object detection are extensively presented. Whereas the suitable metrics for time-series and imbalanced data for LSTM-based DNNs are not addressed.

To address the identified shortcomings, this work investigates

the resilience of LSTM-based DNNs containing convolutional layers. Based on that, we propose two fault-tolerant techniques to mitigate fault effects in parameters for DNNs, with zero overhead in memory. In this paper, we perform resilience analysis on StageNet [22] as a case study in healthcare applications for disease prediction. The contributions of this paper are as follows:

- Performing a comprehensive resilience analysis for various LSTM-based DNNs using fault injection in parameters, leading to observing the effect of different DNN structures and identification of critical bits;
- Proposing two fault-tolerant methods for LSTM-based DNNs, Weights Bit Clipping (WBC) and Activations Value Clipping (AVC), to effectively reduce the impact of faults in parameters on LSTM-based DNNs;
- Demonstrating the efficacy of the proposed methods in DNNs' resilience, leading up to 278.6 times less critical cases in the outputs caused by faulty parameters.

The rest of the paper is organized as follows. Section II provides a background on the LSTM-based DNN structure under study. Section III presents the proposed method for the resilience analysis and improvement of DNNs. Section IV explains the experimental setup and comprehensively demonstrates and analyzes the results. Finally, Section V concludes the paper.

II. Preliminaries on LSTMs and StageNet

Recursive Neural Networks (RNNs) are characterized by recursive loops within their neuronal connections, facilitating the retention of information over time. Long Short-Term Memory (LSTM) networks are introduced to enhance the capability of RNNs for long-term information retention. LSTMs are structured by stacking multiple LSTM layers, each comprising numerous LSTM cells.

StageNet [22] is an LSTM-based DNN designed for effective disease prediction in healthcare applications. It predicts the stage of a patient's disease according to the characteristics of the tests performed by the patient through time. Fig. 1 illustrates the overall structure of StageNet. It is composed of an LSTM layer for characterizing the disease stage over time, a convolutional (CONV) module, and an output Fully Connected (FC) layer to output the predicted disease condition. The CONV module contains one layer in parallel with two FC layers, and their outputs are multiplied point-wise.

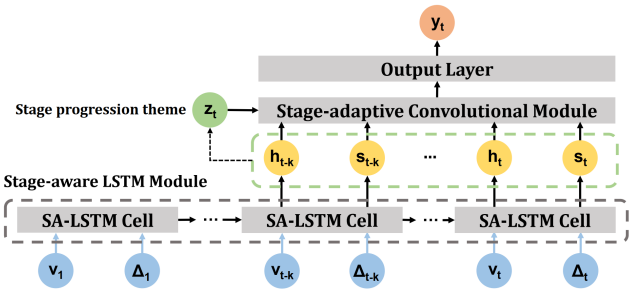


Fig. 1: Overal structure of StageNet [22].

StageNet receives time-series data as inputs according to the patient visits $(v_1, v_2, \cdots, v_t)$ which contain numerical clinical features at different times $(\Delta_1, \Delta_2, \cdots, \Delta_t)$. They pass through the LSTM layer with multiple cells inferring the variation of a patient's health stage considering their current status. The produced results from time-series data by the LSTM layer are forwarded to the CONV module for learning patterns of the disease stages. Afterward, the classification is performed by the output layer for disease stage and risk prediction. In the variations of StageNet, the number of LSTM cells can be customized, and also the presence of the CONV module is arbitrary. In this work, we consider four variations of StageNet for resilience analysis.

To measure the performance of StageNet, the following metrics are evaluated:

- **Accuracy:** This metric represents the percentage of correct predictions of StageNet compared to the expected outputs.
- **AUROC:** This metric refers to the area under the receiver operating characteristic curve illustrating the trade-off between true positive rate and false positive rate. It shows how a classifier can discriminate the positive and negative classes and it is extensively used when a dataset is imbalanced.

Since the distribution of data in the dataset between different disease stages is imbalanced, AUROC is a more suitable metric to show the performance of StageNet. An imbalanced data record is common in medical data, since unstable cases happen less frequently than stable patient conditions. Therefore, for resilience analysis of this LSTM-based DNN, we consider AUROC drop along with accuracy drop as a case study for healthcare applications.

III. Fault Propagation and Mitigation in LSTM-based DNNs

A. Resilience Evaluation Using Fault Injection

As mentioned, DNNs' weights are stored in memory, which is susceptible to soft errors. To assess the potential impact of these threats on the performance of LSTM DNNs, random bits determined by predefined Bit Error Rates (BER) are flipped in the weights of the DNN prior to inference. This process is repeated several times and average results over the repetition are reported.

To analyze the resilience of DNNs comprehensively, random bit flips are applied throughout the DNN's weights to assess the overall network's behavior in the presence of faults. To quantify the resilience analysis, once the average performance metrics (Accuracy and AUROC) for a DNN under test are obtained, their difference with the fault-free metrics is considered as accuracy drop and AUROC drop.

Furthermore, the output effect of faults is categorized into the following classes, to quantify the effect of faults on the DNNs' outputs:

- **Masked:** Outputs remain identical between faulty and fault-free executions.
- **Non-critical Silent Data Corruption (SDC):** Output values differ between faulty and fault-free executions, while the classification result remains consistent.
- **Critical SDC:** Both output values and classification results differ between faulty and fault-free executions.
- **Detected Unrecoverable Error (DUE):** The DNN generates "NaN" values in the output, indicative of a system exception.

Nonetheless, the summation of critical SDCs and DUEs is the total critical cases for a faulty DNN which results in misclassification compared to the fault-free model.

Furthermore, to identify the critical bits in DNNs, we perform a bit-wise fault injection experiment throughout the DNNs. In

such an experiment, one bit is considered as the target and it is flipped in all parameters and the inference is performed and the performance metrics are measured. The bit that has the highest impact on the performance metrics is identified as the most critical bit.

B. Resilience Improvement for LSTM-based DNNs

To enhance the resilience of the LSTM DNNs, first, we profile the bit values in weights and all activation values with validation input data. Observing the bit patterns of weights leads us to the first protection mechanism for memory errors. On the other hand, faulty weights produce erroneous activation values. Therefore, observing the range of values in the activations through a forward pass of the DNN leads us to a second fault tolerance technique for LSTMs. Consequently, we propose two model-level fault tolerance techniques with zero memory overhead: 1) Weights Bit Clipping (WBC), and 2) Activations Value Clipping (AVC).

In the WBC method, all weights of fault-free DNN models are profiled and their bit patterns are analyzed. As a result, a consistent bit pattern is revealed in the DNNs under study. Moreover, using fault injection, the most critical bit is identified. To this end, an extensive exploration of different bit flips is carried out and the resilience is measured for each bit. Therefore, the method suggests clipping the most critical bits to a certain value throughout the DNN, before an inference.

In the AVC method, first, the input values to each activation function of the LSTM cells as well as the CONV and FC layers in DNNs are profiled and their maximum and minimum values are obtained, during a fault-free forward pass with validation data. The obtained values are then utilized for detecting faults that is when an input to corresponding activation functions exceeds the determined value range. Once a fault is detected in a forward pass, the corresponding value is clipped. If an activation value falls below the minimum range value, it is set to that minimum threshold. Conversely, if an activation value exceeds the maximum profiled value, it is set to the maximum threshold.

This technique ensures that activation values remain within a predefined and safe range of values. It is worth mentioning that since LSTM cells possess sigmoid and Tanh activation functions, their outputs are already limited to a certain range (i.e. $[0, 1]$ and $[-1, 1]$ respectively). Therefore, it is challenging to apply AVC to their outputs. Therefore, AVC is applied to the inputs of activation functions throughout the LSTM-based DNNs, whether they are sigmoid, tanh, or ReLU. This method is particularly effective in scenarios where accumulated faults in weights result in producing large erroneous values in the forward pass of a DNN. Therefore, they are clipped to certain values so the error propagation is harnessed.

Ultimately, the effectiveness of WBC and AVC is evaluated by fault injection to determine how they mitigate the fault impact and which technique offers superior fault tolerance in LSTM DNNs. By comparing these methods, we aim to identify the more robust approach for enhancing the reliability of LSTM neural networks in the presence of faults.

IV. Experiments

A. Experimental Setup

To evaluate and improve the resilience of LSTM-based DNNs against faults in parameters, four variations of StageNet are experimented. Two variations represent the full StageNet model which includes CONV layer, with different numbers of LSTM cells (384 and 72), and the two variations that exclude the convolutional module, containing only LSTM layer, also with 384 and 72 LSTM cells.

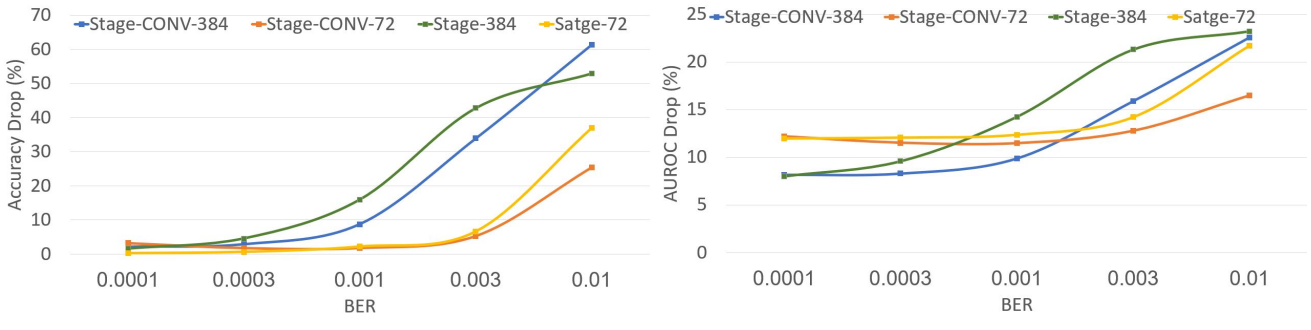
The test data is sourced from the Medical Information Mart for Intensive Care (MIMIC-III) data set, which includes 17 physiological variables recorded at each visit. This data was transformed into a 76-dimensional vector comprising numerical and one-hot encoded categorical clinical features for 33,678 unique patients.

Baseline metrics, including accuracy and AUROC in fault-free executions, are summarized in Table I, alongside the number of parameters for each model. All models were executed on a general-purpose processor supporting 32-bit floating-point IEEE-754 data representation.

Table I: Accuracy, AUROC and the number of different weight sets of the LSTM-based ANNs in this work.

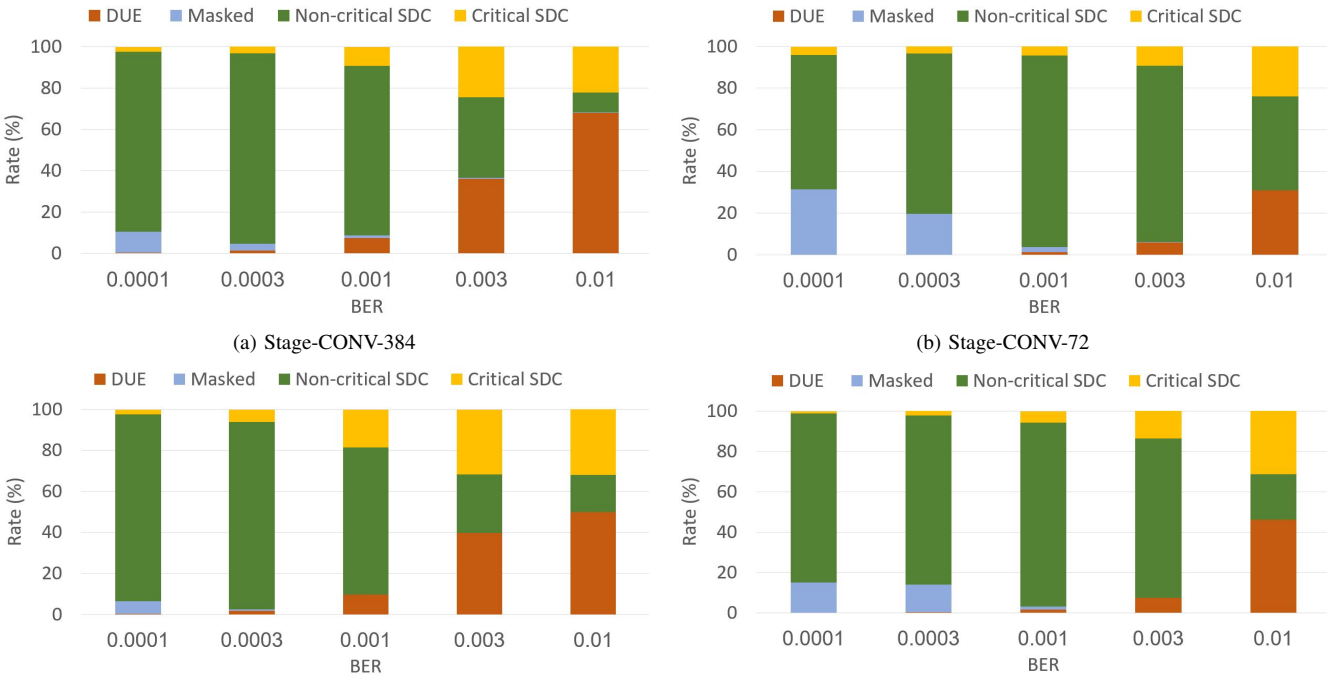
	Accuracy	AUROC	#weights in LSTM	#weights in CONV	#weights in FC
Stage-CONV-384	94.94%	79.21%	738618	442368	24960
Stage-CONV-72	83.75%	76.75%	48764	51840	1800
Stage-384	90.28%	79.29%	738618	0	384
Stage-72	77.28%	76.97%	48764	0	72

We conduct Fault Injection (FI) across all weights in the DNNs under study. The number of injected faults is determined using a Bit Error Rate (BER) ranging from 0.0001 to 0.01, covering a comprehensive range of potential errors. FI is repeated 1000 times to ensure an acceptable confidence level. For each iteration, a drop in accuracy and AUROC is obtained, and faults are classified



(a) (b)

Fig. 2: Model-wise FI results for DNNs based on: a) Accuracy drop, b) AUROC drop.



(a) Stage-CONV-384

(b) Stage-CONV-72

(c) Stage-384

(d) Stage-72

Fig. 3: Fault classification in model-wise FI for a) Stage-CONV-384, b) Stage-CONV-72, c) Stage-384, d) Stage-72.

according to Subsection III-A. Eventually, the average results over all iterations are reported in the paper.

All experiments are implemented and performed using PyTorch and executed on an Intel® Core™ i7-9700 CPU. Through these experiments, we aim to thoroughly assess the reliability of LSTM-based DNNs under faults in parameters and evaluate the effectiveness of the proposed protection techniques in preserving model performance.

B. Resilience Analysis Results

As mentioned, weight parameters across the entire DNN models are the fault space for random bit flips determined by different BERs. As illustrated in Fig. 2a, the performance metrics for all models significantly drop under fault injection campaigns. In Fig 2a, as the BER increases, larger models (i.e., Stage-CONV-384 and Stage-384) demonstrate more accuracy drop than the other models, at the same BERs. However, the AUROC drop metric shows a different behavior. It is observed that Stage-CNN-72 and Stage-72 are remarkably sensitive to faults in terms of their AUROC, at the lowest BER. While at higher BERs, the AUROC drop is higher for larger DNNs.

Regarding Fig. 2b, as AUROC expresses the discrimination of classification over different thresholds, this observation shows that smaller DNNs are remarkably sensitive to faults to correctly distinguish the stage of patients' disease. According to the results, the AUROC metric for all DNNs falls below 60% when $BER = 0.01$. At such high BERs, although larger DNNs are shown to be more error-prone, in this safety-critical application, none of them are reliably functioning. Noteworthy that when AUROC is close to 50% DNNs perform random classification.

On the other hand, it is observed that DNNs possessing the CONV module are generally more resilient than the ones without the CONV module. It shows that the CONV module increases the capability of fault masking in LSTM-based DNNs and improves their inherent resilience.

According to Fig. 3, in all models, as the BER increases, the fault effects with masked and non-critical SDCs decrease, significantly leading to more critical SDCs and DUEs. This figure also evidences the fact that the DNNs with the CONV module are more resilient to faults than the ones without the CONV module. Obtained results indicate that when $BER = 0.01$, total critical SDC and DUE rate for StageNet-CONV-384, StageNet-CONV-72, StageNet-384 and StageNet-72 is 90.05%, 54.97%, 82.02% and 77.33%.

These findings highlight the importance of fault mitigation mechanisms, to enhance the fault tolerance and overall reliability of LSTM-based DNNs. To this end, we perform a bit-level analysis of weights. First, we observe the average value of each bit throughout the weights in all DNNs, as illustrated in Fig. 4. It is observed that bits 0 to 26 in 32-bit floating point data representation almost have a unified distribution between 0 and 1. while bits number 27, 28, and 29 are always '1' and bit number 30 is always '0'. It shows that we can protect the bits that are constant throughout the weights.

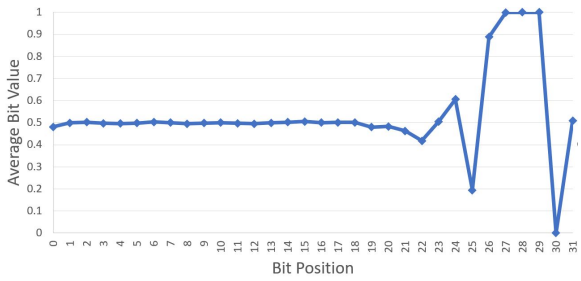


Fig. 4: Average value of different bits.

However, to decrease the fault-tolerance overhead, we further

analyze the resilience of DNNs against specific bit positions, as described in Section III. As depicted in Fig. 5, the accuracy drop for bit 30 in all DNNs is significantly higher than bit-flip in other bits. Consequently, bit 30 is identified as the most critical bit in the DNNs, which is observed in Fig. 4 that its value is always '0'. As a protection mechanism, this bit is always set to '0', which ensures that the most critical bit is consistently safeguarded, enhancing the overall reliability of the LSTM-based DNNs.

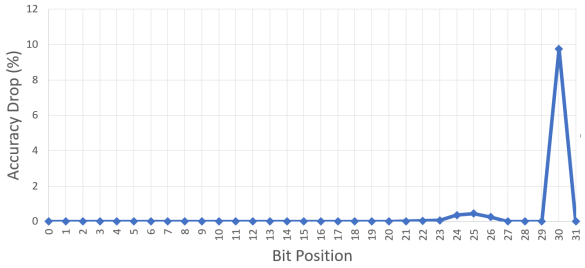


Fig. 5: Fault injection results for finding the most critical bit of weight parameters.

C. Fault Tolerance for LSTM-based DNNs

1) Weights Bit Clipping (WBC)

As shown in Fig. 6, weights bit clipping lays a significant improvement on the reliability of LSTM-based DNNs. According to Fig. 6a, the accuracy drop is close to zero through all BERs for all protected models. On the other hand, the AUROC drop in Fig. 6b is remarkably improved for all DNNs. According to the results, the AUROC drop is reduced by up to 3.2x when $BER = 0.01$.

According to the fault classification results, the critical SDC and DUE rate is remarkably reduced in the protected DNNs by the WBC method. The protection mechanism is capable of effectively removing all DUE impacts on the outputs, as it does not allow producing any large value by clipping the MSB in weights. As a result, the total critical SDC and DUE rate across DNNs is reduced by up to 278.6x.

2) Activations Value Clipping (AVC)

Fig. 7 presents how effectively activations value clipping protects the models against faults. According to the results, The accuracy drop and AUROC drop are reduced by up to 15.54x and 1.5x among the DNNs, respectively, when $BER = 0.01$. The fault classification results through the fault injection campaigns on protected DNNs by AVC indicate that this method is also capable of removing all DUE effects. As a result, the total DUE and critical SDC rate for StageNet-CONV-384, StageNet-CONV-72, StageNet-384 and StageNet-72 is 13.88%, 5.18%, 36.47%, and 16.86% resulting in up to 10.6x reduction across DNNs when $BER = 0.01$.

3) Comparison

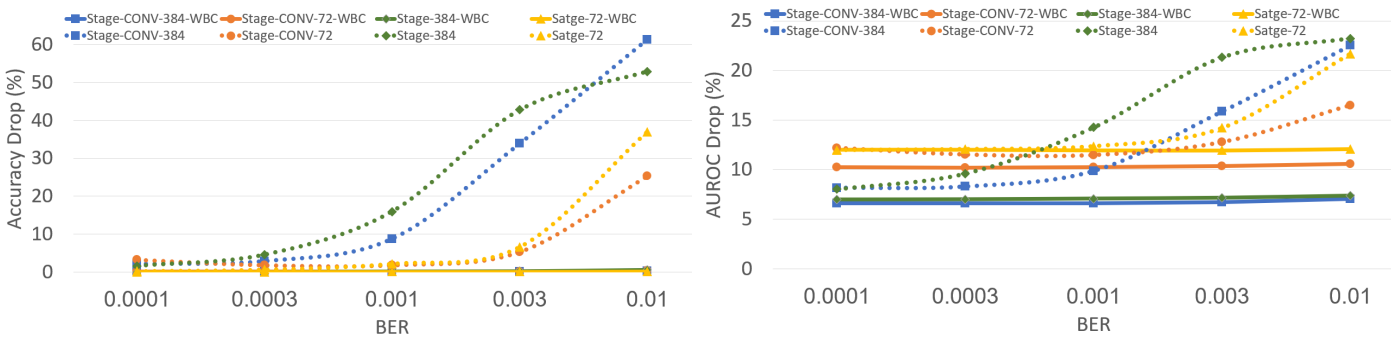
Fig. 8 compares the proposed zero-memory overhead fault-tolerant techniques for LSTM-based DNNs. As observed, Weights Bit Clipping (WBC) generally demonstrates a more consistent protective effect across different DNNs. It effectively reduces accuracy drop and the incidence of critical faults across the board. However, Activations Value Clipping (AVC) slightly outperforms in a few cases. WBC achieves 2.36x, 1.19x and 2.26x less AUROC drop than AVC in Stage-CONV-384, Stage-CONV-72, and Stage-384, when $BER = 0.01$, whereas AVC provides 1.13x times less AUROC drop than WBC for Stage-72 at the same BER.

Nonetheless, each proposed fault-tolerance technique is applicable in different design scenarios. WBC applies directly to the memory and can be conducted to the bit values of the stored data before an inference. On the other hand, AVC is applied during the inference and prevents errors produced by faulty weights during the inference.

V. Conclusion

In this paper, we study the reliability of various LSTM-based DNNs (variants of StageNet) in healthcare as a case study with different structures and propose two zero-memory overhead fault-tolerance techniques for them. Using fault injection, we analyzed the resilience of different structures with and without convolutional layers within DNNs. Results indicate that LSTM-based DNNs possessing convolutional layers demonstrate more resilience than the ones without convolutional layers. Furthermore, we performed bit-level analysis resulting in the identification of the most critical bits.

Moreover, two zero-overhead protection techniques to improve their fault tolerance are proposed: weights bit clipping and activations value clipping. Through comprehensive fault injection experiments on variations of StageNet, we evaluate the effectiveness of these techniques in mitigating the impact of faults. It is shown that weights bit clipping can reduce the AUROC drop by up to 3.2x and DUE and critical faults by up to 278.6x compared to the unprotected DNNs at a high BER. Also, activations value clipping reduces the AUROC drop by 1.5x and DUE and critical SDCs by 10.6x, under the same conditions.



(a) (b)

Fig. 6: Weights Bit Clipping protection and FI results for a) Accuracy drop, b) AUROC drop.

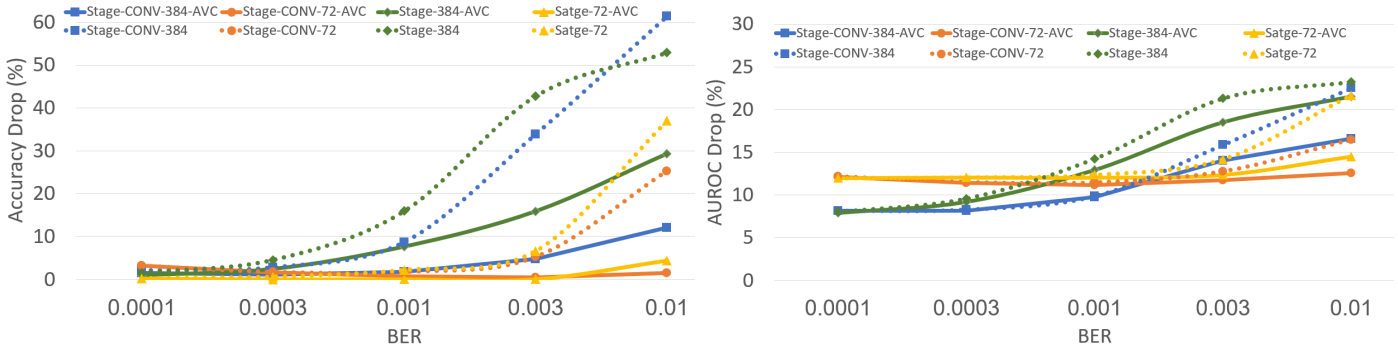


Fig. 7: Activations Value Clipping protection and FI results for a) Accuracy drop, b) AUROC drop.

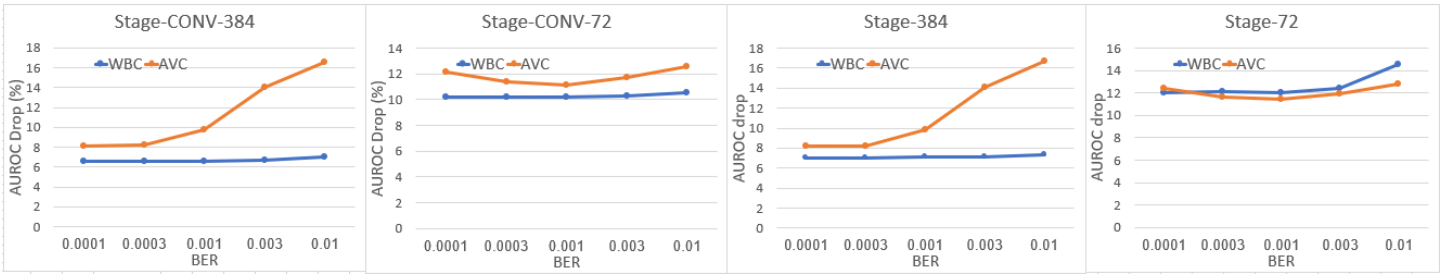


Fig. 8: Weights Bit Clipping (WBC) and Activations Value Clipping (AVC) comparison based on AUROC drop.

Thus, the results demonstrate that the weights bit clipping method is extremely effective in mitigating the effect of faults occurring in the parameters of LSTM-based DNNs.

References

[1] M. Loni, H. Mousavi, M. Riazati, M. Daneshtalab, and M. Sjödin, "Tas: ternarized neural architecture search for resource-constrained edge devices," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 1115–1118.

[2] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," *Computational intelligence and neuroscience*, vol. 2018, 2018.

[3] B. Rokh, A. Azarpeyvand, and A. Khanteymoori, "A comprehensive survey on model quantization for deep neural networks in image classification," *ACM Transactions on Intelligent Systems and Technology*, vol. 14, no. 6, pp. 1–50, 2023.

[4] M. H. Ahmadilivani, M. Taheri, J. Raik, M. Daneshtalab, and M. Jenihhin, "A systematic literature review on hardware reliability assessment methods for deep neural networks," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–39, 2024.

[5] F. Su, C. Liu, and H.-G. Stratigopoulos, "Testability and dependability of ai hardware: Survey, trends, challenges, and perspectives," *IEEE Design & Test*, 2023.

[6] Y. Ibrahim, H. Wang, J. Liu, J. Wei, L. Chen, P. Rech, K. Adam, and G. Guo, "Soft errors in dnn accelerators: A comprehensive review," *Microelectronics Reliability*, vol. 115, p. 113969, 2020.

[7] A. Azizimazreah, Y. Gu, X. Gu, and L. Chen, "Tolerating soft errors in deep learning accelerators with reliable on-chip memory designs," in *2018 IEEE International Conference on Networking, Architecture and Storage (NAS)*. IEEE, 2018, pp. 1–10.

[8] M. A. Neggaz *et al.*, "Are cnns reliable enough for critical applications? an exploratory study," *IEEE Design & Test*, vol. 37, no. 2, pp. 76–83, 2019.

[9] R. Manne and S. C. Kantheti, "Application of artificial intelligence in healthcare: chances and challenges," *Current Journal of Applied Science and Technology*, vol. 40, no. 6, pp. 78–89, 2021.

[10] E. J. Harris *et al.*, "A survey of human gait-based artificial intelligence applications," *Frontiers in Robotics and AI*, vol. 8, p. 749274, 2022.

[11] V. Sze *et al.*, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[12] B. Lim and S. Zohren, "Time-series forecasting with deep learning: a survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, pp. 202–209, 2021.

[13] K. Adam *et al.*, "A selective mitigation technique of soft errors for dnn models used in healthcare applications: Densenet201 case study," *IEEE Access*, vol. 9, pp. 65 803–65 823, 2021.

[14] C. Liu, C. Chu, D. Xu, Y. Wang, Q. Wang, H. Li, X. Li, and K.-T. Cheng, "Hyca: A hybrid computing architecture for fault-tolerant deep learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 10, pp. 3400–3413, 2021.

[15] Y. Zhao, K. Wang, and A. Louri, "Fsa: An efficient fault-tolerant systolic array-based dnn accelerator architecture," in *2022 IEEE 40th International Conference on Computer Design (ICCD)*. IEEE, 2022, pp. 545–552.

[16] M. H. Ahmadilivani *et al.*, "Enhancing fault resilience of qnns by selective neuron splitting," in *2023 IEEE 5th AICAS*, 2023, pp. 1–5.

[17] L.-H. Hoang *et al.*, "Ft-clipact: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation," in *DATE*, 2020, pp. 1241–1246.

[18] B. Ghavami *et al.*, "Fitact: Error resilient deep neural networks via fine-grained post-trainable activation functions," in *2022 DATE*. IEEE, 2022, pp. 1239–1244.

[19] M. H. Ahmadilivani, S. Mousavi, J. Raik, M. Daneshtalab, and M. Jenihhin, "Cost-effective fault tolerance for cnns using parameter vulnerability based hardening and pruning," in *2024 IEEE IOLTS, inpress*, 2024, pp. 1–7.

[20] N. Nosrati and Z. Navabi, "Analysis and enhancement of resilience for lstm accelerators using residue-based ceds," *IEEE Access*, 2024.

[21] M. H. Ahmadilivani, J. Raik, M. Daneshtalab, and A. Kuusik, "Analysis and improvement of resilience for long short-term memory neural networks," in *2023 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. IEEE, 2023, pp. 1–4.

[22] J. Gao, C. Xiao, Y. Wang, W. Tang, L. M. Glass, and J. Sun, "Stagenet: Stage-aware neural networks for health risk prediction," in *Proceedings of The Web Conference 2020*, 2020, pp. 530–540.