# Nadam Bound: A Novel Approach Of Subsuming Nesterov Momentum And Dynamic Bounding Into Adaptive Moment Estimation To Enhance The Detection Accuracy Of Deep Learning In Real World Steganalysis

**V.Hema Malini[1] , C.Victoria Priscilla[2]**

[1]Research Scholar, Shrimathi Devkunvar Nanalal Bhatt Vaishnav College for Women, University of Madras, Chennai, India
[2]Associate Professor, Shrimathi Devkunvar Nanalal Bhatt Vaishnav College for Women, University of Madras, Chennai, India

Received: 17.07.2024 Revised: 18.08.2024 Accepted: 20.09.2024

**ABSTRACT**
Within the realm of information security, steganography and stegan analysis are related concepts. The goal of steganalysis is to find hidden data in digital media. Steganographic techniques are always developing, leading to ongoing changes in steganalysis based on CNN.

An optimizer in CNN tunes the weights while training, to reduce errors and boost the efficiency of the model. It is a key for any deep neural network learning to be both successful and efficient. Steganalysis based on deep learning is not an exception to the rule. Various deep network types usually require different optimizers, which must be selected through several experiments. To enhance the model's training speed effectively across deep networks, NAdamBound approach is introduced, a hybrid optimizer that combines Nadam's Nesterov momentum technique with Adabound's dynamic bounding mechanism. This could give rise to both the rapid convergence qualities from Nesterov momentum and the adjustable learning rate benefits.Extensive experiments conducted on Steganalysis of real-world dataset proved NAdamBound produced 94%, 92.2% and 93.8% accuracy against WOW, S-UNIWARD and HILL which surpasses its corresponding fellow optimizers in reducing loss and boosting accuracy.

**Keywords:** Steganalysis, Deep learning, Optimizer, Adam, AdaBound, Convolution Neural Network, Cyber Security

## INTRODUCTION

With the goal of concealing sensitive information from prying eyes and preventing observable distortions and modifications to statistical properties, steganography is a technology that enables two parties to communicate covertly. Any digital media can be thought of as a carrier, but since digital images are communicated via the Internet in bulk, they are among the most widely used carriers. As a countermeasure to image steganography, image steganalysis looks for hidden data that has been implanted using a steganography technique. In their conflict and evolution, image steganography and image steganalysis[1] are rivals. After the advent of deep learning, many new architectures were introduced using CNN for effective detection in image steganalysis. The architectures use different types of CNN in feature extraction for achieving good accuracy.

In order to accomplish a fast training procedure with an element-wise scaling term on learning rates, adaptive optimization techniques have been presented. For example, Adagrad[2], RMSProp[3], Adam[4]. However, the generalization is still less than that of SGD. They break to converge because of high learning rates and instability. Adabound [5] addressed this problem by placing dynamic bounds on these learning rates to guarantee that the convergence occurs to a specific range over time. Additionally, adding Nesterov momentum[6] to Adam (Nadam) could speed up convergence. Therefore, combining Adabound's dynamic learning rate bounds with Nadam's[7] momentum changes will speed up convergence, generalize, decrease loss, and ultimately increase accuracy. The advantages of Adabound, Nesterov Accelerated momentum, and Adam are combined in this work to present a novelNAdamBound approach. Furthermore, a lot of earlier studies employed pre-existing databases[8] for their investigations. This study's focus on a wide range of applications is made possible by the utilization of real-world data sets.

## RELATED WORK

Deep learning relies on optimization, which is essential to the performance of neural network models in a variety of applications. During training, it adjusts the parameters of a neural network. Its main aim is to improve performance by reducing the error or loss function of the model.

In many scientific domains, Stochastic gradient descent-based optimization is fundamental. Every iteration process randomly selects batches of data rather than the complete dataset, enabling deeper learning model optimization that is both computationally viable and more efficient. The learning rate and initial parameters are chosen. The data is randomly jumbled in each iteration in order to approach a minimum. SGD takes longer to get to the local minima in terms of iterations. The calculation cost remains low even when the number of repetitions is increased. SGD might not converge to the precise global minimum and might produce a suboptimal solution as a result of the noisy updates.

Adagrad employs varying learning rates for every iteration. Learning rate decreases with increasing parameter modification. Real-world datasets contain both dense and sparse features, therefore this change is really helpful. Adagrad has the advantage of eliminating the requirement for manual learning rate modification. The learning rate may eventually drop to a very low level. Small learning rates ultimately stop the model from learning new information, which compromises the model's accuracy.

RMSProp mostly concentrates on quickening the optimization process to attain the local minimum by reducing the quantity of function evaluations applied. Compared to gradient descent algorithms and its variations, it converges faster and requires less adjustment. The issue with RMSProp is that not all applications can use the recommended value, and the learning rate must be explicitly defined. There is also no bias-correction phrase in RMSProp. When the bias is not corrected, excessively large step sizes and frequently divergence may occur.

Adaptive learning is the backbone of Adadelta [9], which addresses major issues with Adagrad and RMSProp optimizer. It deals with problems pertaining to historical gradient formation and learning rate tuning. Adadeltaimproves convergence and stability during training by dynamically adjusting the learning rate in response to accumulated gradient information.

The Adamoptimizer, also known as the Adaptive Moment Estimation optimizer, is a Stochastic Gradient Descent (SGD) extender. Individual learning rates are dynamically computed by theAdamoptimizer using the second moments of the previous gradients. During training, the Adam optimizer reaches an adaptive learning rate that allows it to effectively traverse the optimization space. This adaptivity aids in the neural network's quicker convergence and enhanced performance.

Similar to Adam, Adabound adjusts learning rates depending on gradients, but it also adds dynamic bounds to these rates to guarantee that they eventually converge to a specific range. By keeping learning rates steady, it aids in improving generalization. By adding Nesterov momentum, Nadam improves on Adam by projecting future values of the parameter space and modifying gradients appropriately.

With bounded learning rates of Adabound, the current study provides better generalization, even to unknown data, and faster convergence with Nesterov sped up Adaptive Moment, or Adam.

Algorithm 1:Subsuming Nesterov momentum and Adaptive Bounding into ADAptive Moment estimationfor Stochastic optimization. (NAdamBound)

**Default settings:**

Initial learning rate $\alpha = 0.001$;
momentum for the first moment $\beta1 = 0.9$;
momentum for the second moment$\beta2 = 0.999$;
To prevent division by zero$\varepsilon = 10^{-8}$;
lower bound for learning rateη_lower=1e-5;
upper bound for learning rateη_upper=1

**Requisite:**

α: Step size;
$\beta1, \beta2 \in [0, 1)$: Exponential decay rates for moment estimates;
$f(\theta)$: Stochastic objective scalar function;
$\theta_0$: Initial parameter vector;
η_lower, η_upper: Lower and upper bound for learning rate clipping;

```
$m_0$ , $n_0$ , $t \leftarrow 0$          (initialize 1st moment vector, 2nd moment vector, time step)
**While**$\theta_t$ not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$          (Obtain Gradients w.r.t. stochastic objective at timestep t)
```

```
        $m_t \leftarrow \beta1 \cdot m_{t-1} + (1 - \beta1) \cdot g_t$   (Upgrade biased 1st moment estimate)
        $n_t \leftarrow \beta2 \cdot n_{t-1} + (1 - \beta2) \cdot {g_t}^2$   (Upgrade biased 2nd raw moment estimate)
        $\hat{m}_t \leftarrow m_t / (1 - \beta1^t)$   (Evaluate bias-corrected 1st moment estimate)
        $\hat{n}_t \leftarrow n_t / (1 - \beta2^t)$   (Evaluate bias-corrected 2nd raw moment estimate)
        $\tilde{m}_t \leftarrow \frac{\beta1 \cdot m_t}{1-\beta1^t} + \frac{(1-\beta1) \cdot g_t}{1-\beta1^t}$   (Evaluate Nesterov-accelerated gradient)
        $\eta\_lower_t \leftarrow \eta\_lower \cdot (1 - 1 / (1 + \gamma \cdot t))$
        $\eta\_upper_t \leftarrow \eta\_upper \cdot (1 + 1 / (1 + \gamma \cdot t))$   (dynamic learning rate bounds)
        $\eta_t \leftarrow clip\ (\alpha / (\sqrt{\hat{n}_t} + \varepsilon), \eta\_lower_t, \eta\_upper_t)$ (Evaluate and clip the effective learning rate)
        $\theta_t \leftarrow \theta_{t-1} - \eta_t \cdot \tilde{m}_t$   (Upgrade parameters)
end While
return $\theta_t$   (derived parameter)
```

**Steps involved**

1. Initialize first and second moment vector and time step
2. Gradient with respect to the stochastic objective is obtained
3. First and second moment estimates calculated
4. Bias correction applied
5. Nesterov momentum used for updating parameters
6. Dynamic learning rate bounds calculated
7. Learning rate clipped within bounds
8. Model parameters updated

NAdamBound pseudocode is given in Algorithm 1. f(θ) is the objective scalar function differentiable with respect to θ (parameters). The initial parameter vector is $\theta_0$. The initialisation of the moving averages $m_t$ and $n_t$ is done as (vectors of) 0's. The stochastic function is realized at successive time steps(t←t+1). The gradient is denoted with $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ evaluated at timestep t (partial derivative of $f_t$ with respect to θ). The exponential moving averages of the gradient and squared gradient ($m_t$ and $n_t$) i.e., the first and second moments of the gradients are updated. The exponential decay rates β1, β2 ∈ [0, 1] are controlled by the hyper parameters.

Since the moving averages $m_t$ and $n_t$ are initialized to 0, the moment estimates of the bias tend to zero in the early stages. Bias correction is done for the first moment estimate ($\hat{m}_t$) and second raw moment estimate ($\hat{n}_t$). Nesterov's accelerated gradient (NAG)[6] can be reformulated as a type of improved momentum. NAG has an evidently better bound over gradient descent on convex and non-stochastic objectives. The initial moment adjustment is supplemented with the Nesterov term, which combines momentum with a forward-looking "lookahead" derived from the present gradient denoted by $\tilde{m}$ as inNADAM.

The learning rate $\eta_t$ is constrained adaptively within lower $\eta\_lower_t$ and upper $\eta_upper_t$bounds as in ADABOUND. This prevents the learning rate from drastically decreasing or increasing over time. To prevent gradient explosion, gradients on learning rates higher than a threshold are trimmed, which is an approach inspired by gradient clipping. The behaviour is initially similar to ADAM then gradually changes to SGD as the constraints get more and more limited because the bounds have little effect on learning rates. The bounded learning rate performs parameter updating, which contributes to training stability, particularly as convergence drops down approaching the end of optimization.

**Experiment**

An optimizer is an essential component in deep learning that calibrates the parameters of a neural network while training. Its main aim is to improve performance by reducing the error or loss function of the model. In this study, the NAdam Bound optimizer is employed to improve accuracy, accelerate up convergence, enhance generalization and reduce loss. Four optimization algorithms are tested: Adam, Nadam, Adabound and NAdamBound.

The DDS_SE-Net architecture[10], that performs well on real-world datasets, is the one employed in the present research. In its feature extraction stage, DDS_SE-Net makes use of the Dilation[11], Depthwise Separable Convolution[12], Squeeze and Excitation blocks[13]. In order to avoid over fitting and save computing costs, separable convolutions incorporate both depth-wise and point-wise convolutions. Dilations lower the processing cost by assisting in the detection of characteristics at different scales. SE blocks[14]are introduced in order to adaptively balance the channels.

The analyses were carried out using a real-world dataset made up of 5000 photos that were gathered online (Cover). Three spatial steganographic algorithms, WOW[15], S-UNIWARD[16]and HILL[17] and with payloads of 0.4bpp and 0.2bpp, were used to produce stego visuals. The findings were summarized using 4000 image pairings for training and 1000 pairs for testing. The stego pictures generated, together with the cover images, were loaded into the DDS_SE-Net architecture, and the outcomes were recorded.

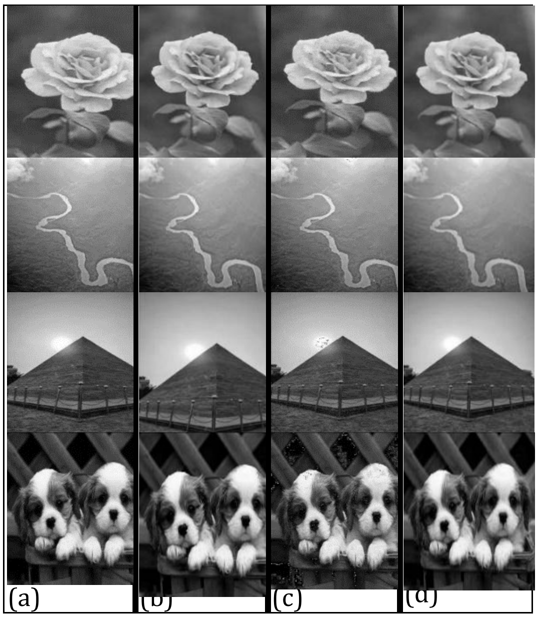


**Figure 1.** Image samples of (a) Cover (b) Stego with WOW 0.4bpp (c) Stego with S-Uniward 0.4bpp (d) Stego with HILL 0.4bpp

## RESULTS AND DISCUSSION

For the WOW, S-UNIWARD, and HILL steganography algorithms, Table 1 shows the results of the DDS_SE-Net. Payloads of 0.4 and 0.2 bpp were employed as a comparison. The NAdamBound optimizer's results are contrasted with those of the Adam, Adabound, and Nadam optimizers. Figure displays a comparison of the accuracy of the suggested optimizer with three other optimizers.

From Table 1 it could be noted that the accuracy of all stochastic optimization algorithms is equally good with slight changes. The test results are tabulated and it could be seen that all optimization algorithms perform well against WOW steganographic algorithm than HILL and S-UNIWARD. Adam or adaptive moment estimation, is a computationally advantageous method that works well with huge data and/or parameter sets. It gave 92.9% accuracy with 0.4bpp payload and 93.2% accuracy with 0.2bpp payload against WOW. Adabound performance is enhanced evidently with approximately 0.43% improvement in the test accuracy than that of Adam against WOW with 0.4bpp. Accuracy of Nadam is more or less equal to that of Adam with both payloads. Novel NAdamBound optimizer produced 94% and 93.2% with 0.4 and 0.2bpp against WOW which is 1.18% increase than Adam, 0.75% more than AdaBound and 1.18% more than Nadam optimizer bpp being 0.4.

**Table 1.** Comparison of Evaluation Metrics of NAdamBound with Adam, AdaBound and Nadam using DDS_SE-Net against WOW, S-UNIWARD, and HILL with 0.4bpp and 0.2bpp

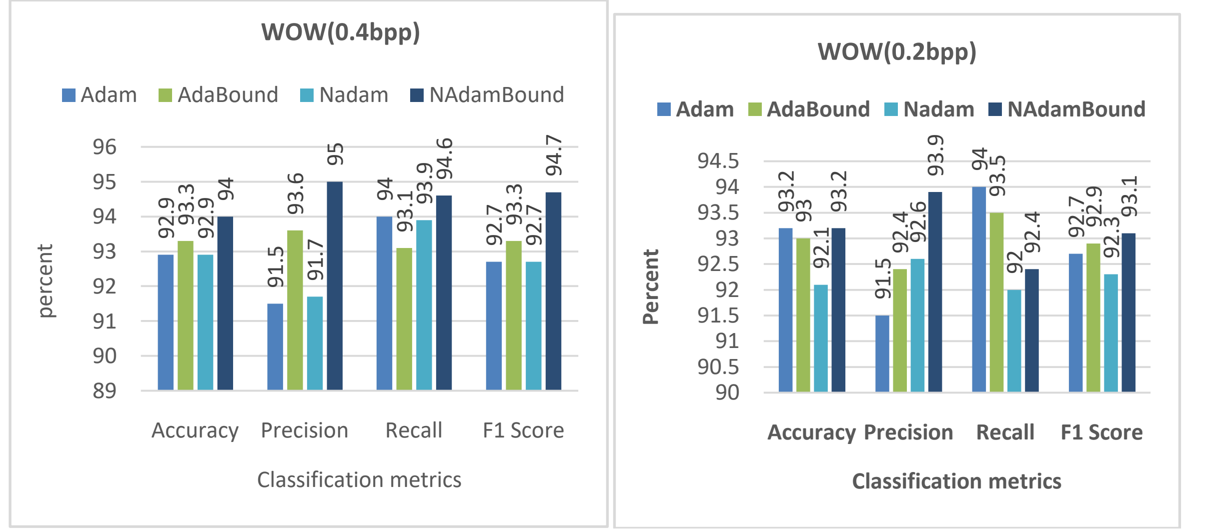
| Steganographic algorithm | Optimizer | Payload | Accuracy | Precision | Recall | F1 Score | Loss |
|---|---|---|---|---|---|---|---|
| WOW | Adam | 0.4 | 92.9 | 91.5 | 94.0 | 92.7 | 21.0 |
| | | 0.2 | 93.2 | 91.5 | 94.0 | 92.7 | 21.0 |
| | AdaBound | 0.4 | 93.3 | 93.6 | 93.1 | 93.3 | 21.0 |
| | | 0.2 | 93.0 | 92.4 | 93.5 | 92.9 | 21.2 |
| | Nadam | 0.4 | 92.9 | 91.7 | 93.9 | 92.7 | 21.4 |
| | | 0.2 | 92.1 | 92.6 | 92.0 | 92.3 | 23.0 |
| | NAdamBound | 0.4 | 94.0 | 95.0 | 94.6 | 94.7 | 20.0 |
| | | 0.2 | 93.2 | 93.9 | 92.4 | 93.1 | 19.2 |
| S-UNIWARD | Adam | 0.4 | 89.2 | 91.4 | 86.9 | 89.0 | 31.9 |
| | | 0.2 | 89.0 | 90.4 | 87.9 | 89.1 | 30.0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | AdaBound | 0.4 | 91.4 | 90.0 | 91.8 | 90.9 | 26.4 |
| | | 0.2 | 91.2 | 90.4 | 91.6 | 91.4 | 25.2 |
| | Nadam | 0.4 | 88.0 | 86.9 | 89.0 | 87.9 | 24.3 |
| | | 0.2 | 91.3 | 92.0 | 92.8 | 92.4 | 20.1 |
| | NAdamBound | 0.4 | 92.2 | 91.7 | 92.0 | 91.8 | 19.6 |
| | | 0.2 | 92.1 | 93.3 | 90.9 | 92.1 | 19.8 |
| HILL | Adam | 0.4 | 89.8 | 91.5 | 87.9 | 89.6 | 30.3 |
| | | 0.2 | 90.7 | 91.9 | 90.4 | 91.1 | 18.0 |
| | AdaBound | 0.4 | 92.0 | 90.9 | 92.8 | 91.8 | 18.4 |
| | | 0.2 | 91.0 | 91.3 | 91.6 | 91.4 | 18.2 |
| | Nadam | 0.4 | 92.3 | 91.8 | 93.2 | 92.5 | 19.2 |
| | | 0.2 | 91.0 | 91.7 | 90.9 | 91.3 | 19.4 |
| | NAdamBound | 0.4 | 93.8 | 93.5 | 94.6 | 94 | 18.0 |
| | | 0.2 | 92.4 | 93.2 | 92.9 | 93 | 17.8 |

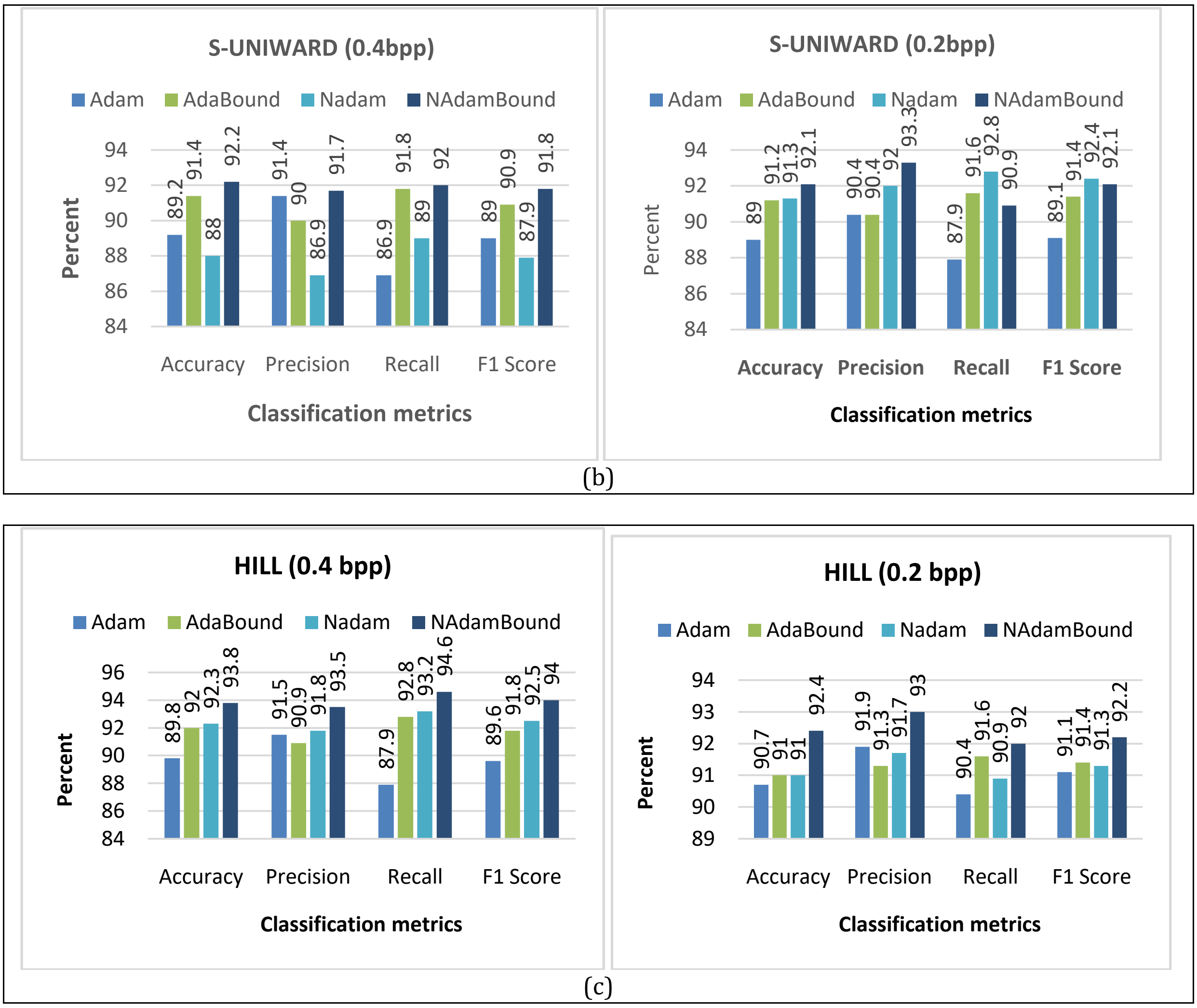
Against S-UNIWARD, Adam produced an accuracy of 89.2% and 89% with 0.4bpp and 0.2bpp respectively. Whereas Adabound produced 91.4% and 91.2 % with 0.4 and 0.2bpp which is an increase of 2.47% than Adam. Nadam produced 1.3% decrease with 0.4bpp and 2.5% increase with 0.2bpp.NAdamBound optimizer produced 92.2% and 92.1% with 0.4 and 0.2bpp against S-UNIWARD which is 3.25% and 3.3% increase than Adam, 0.9% and 0.98% more than AdaBound and 4.5% and 0.87% more than Nadam optimizer with 0.4 and 0.2 bpp respectively.

Against HILL, Adam produced 89.8% and 90.7% accuracy with 0.4bpp and 0.2bpp respectively. Adabound produced 92% and 91 % with 0.4 and 0.2bpp which is an increase of 2.4% with 0.4bpp than Adam. Nadam produced 2.7% increase with 0.4bpp.NAdamBound optimizer produced 93.8% and 92.4% with 0.4 and 0.2bpp against S-UNIWARD which is 4.2% and 1.8% increase than Adam, 1.9% and 1.5% more than AdaBound and 1.6% and 1.5% more than Nadam optimizer with 0.4 and 0.2 bpp respectively.

With the three parent algorithms, Adam, Nadam, and AdaBound, the proposed NAdamBound algorithm is distinct. The key distinction here is that, unlike the previous studies, which relied on pre-existing steganography datasets for their research, the present study's investigations were conducted using real-world data, providing superior generality. The displayed findings are from testing. Figure 2 displays a comparison of the measures such as Accuracy, Precision, Recall, and F1-score with respect to the other optimization techniques. Using real-world datasets, NAdamBound outperforms the other methods in terms of accuracy. The reduction in loss employing the current study in comparison to earlier researches is shown in Figure 3.



(a)



(b)

(c)

**Figure 2.** Comparison of Evaluation Metrics of NAdamBound with Adam, AdaBound and Nadam using DDS_SE-Net against (a) WOW, (b) S-UNIWARD, and (c )HILL with 0.4bpp and 0.2bpp

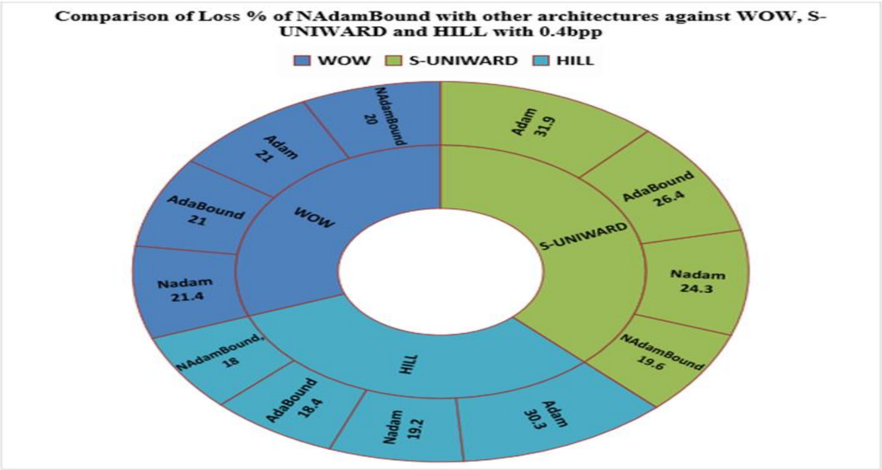


**Figure 3.** Comparison of Loss% of NAdamBound with other architectures against WOW, S-UNIWARD and HILL with 0.4bpp

## CONCLUSION

A significant concept in cyber security is Steganalysis. Steganalysis based on deep learning has gained popularity since it handles large amounts of data and produces remarkable outcomes. In order to facilitate a smooth and progressive transition from adaptive techniques to SGD, this paper proposes a

Novel NAdamBound optimizer for steganalysis that uses dynamic limits on learning rates preventing oscillations in them and integrates Nesterov momentum into adaptive moment estimation (Adam) which aids in accelerated convergence. This facilitates a possible quicker optimization process with a gradually decreasing loss, which raises accuracy.Comparing with parent optimization techniques, NAdamBound gave around 1% to 2% increase in accuracy overall against various spatial steganographic algorithms and payloads. This study's strong generalization capacity is further demonstrated by the use of datasets gathered from actual real-world situations. The need for first order gradients also results in a decrease in total memory use. Additionally, the DDS_SE-Net design minimizes computing costs and time consumption by preventing overfitting. Future studies will focus on other dataset kinds and deep models, paving the path for advancements in Steganalysis.

## REFERENCES

[1] C. V. Priscilla and V. H. Malini, “Steganalysis Techniques: A Systematic Review,” Remit. Rev., vol. 7, no. 1, pp. 171–194, 2022, doi: 10.47059/rr.v7i1.2405.

[2] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” in Proceedings of the IEEE Conference on Decision and Control, 2012, vol. 12, pp. 5442–5444. doi: 10.1109/CDC.2012.6426698.

[3] G. Tieleman, T. and Hinton, “Divide the Gradient by a Running Average of Its Recent Magnitude.,” coursera: Neural Networks for Machine Learning. pp. 4,26-30, 2012. [Online]. Available: coursera: Neural Networks for Machine Learning

[4] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–15, 2015.

[5] L. Luo, “ADAPTIVE GRADIENT METHODS WITH DYNAMIC BOUND OF LEARNING RATE,” Conf. Pap. ICLR, no. 2018, pp. 1–19, 2019.

[6] Y. E. Nesterov, “Nesterov.Pdf,” Soviet Math. Dokl., vol. 27, no. 2. pp. 372–376, 1983.

[7] T. Dozat, “Incorporating Nesterov Momentum into Adam,” ICLR Work., no. 1, pp. 2013–2016, 2016.

[8] A. Westfeld and A. Pfitzmann, “Attacks on steganographic systems breaking the steganographic utilities ezstego, jsteg, steganos, and s-tools–and some lessons learned,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 1768, pp. 61–76, 2000, doi: 10.1007/10719724_5.

[9] M. D. Zeiler, “ADADELTA: An Adaptive Learning Rate Method,” 2012, [Online]. Available: http://arxiv.org/abs/1212.5701

[10] V. H. C Victoria Priscilla, “A Three-Component Feature Extraction Using DDS _ SE-NET for Efficient Deep Learning-Based Image Steganalysis for Real-World Images,” INDIAN J. Sci. Technol., pp. 3335–3343, 2024.

[11] X. Lei, H. Pan, and X. Huang, “A dilated cnn model for image classification,” IEEE Access, vol. 7, pp. 124087–124095, 2019, doi: 10.1109/ACCESS.2019.2927169.

[12] R. Zhang, F. Zhu, J. Liu, and G. Liu, “Depth-Wise Separable Convolutions and Multi-Level Pooling for an Efficient Spatial CNN-Based Steganalysis,” IEEE Trans. Inf. Forensics Secur., vol. 15, pp. 1138–1150, 2020, doi: 10.1109/TIFS.2019.2936913.

[13] J. Hu, “Squeeze-and-Excitation_Networks,” Cvpr, pp. 7132–7141, 2018, [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Hu_Squeeze-and-Excitation_Networks_CVPR_2018_paper.html

[14] G. Xu, Y. Xu, S. Zhang, and X. Xie, “SFRNet: Feature Extraction-Fusion Steganalysis Network Based on Squeeze-and-Excitation Block and RepVgg Block,” Secur. Commun. Networks, vol. 2021, 2021, doi: 10.1155/2021/3676720.

[15] S. Binghamton, “Designing Steganographic Distortion Using Directional Filters,” Ieeexplore.Ieee.Org, pp. 234–239, 2012, [Online]. Available: http://ieeexplore.ieee.org/abstract/document/6412655/

[16] V. Holub, J. Fridrich, and T. Denemark, “Universal distortion function for steganography in an arbitrary domain,” Eurasip J. Inf. Secur., vol. 2014, pp. 1–13, 2014, doi: 10.1186/1687-417X-2014-1.

[17] B. Li, M. Wang, J. Huang, and X. Li, “A NEW COST FUNCTION FOR SPATIAL IMAGE STEGANOGRAPHY College of Information Engineering , Shenzhen University , Shenzhen , GD 518060 , China Institute of Computer Science and Technology , Peking University , Beijing 100871 , China,” Int. Conf. Image Process., pp. 4206–4210, 2014.