IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 9 May 2024, accepted 26 May 2024, date of publication 30 May 2024, date of current version 11 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3407546

RESEARCH ARTICLE

AdamW+: Machine Learning Framework to Detect Domain Generation Algorithms for Malware

AWAIS JAVED[1], IMRAN RASHID[1], SHAHZAIB TAHIR[1], (Senior Member, IEEE), SAQIB SAEED[2], (Senior Member, IEEE), ABDULLAH M. ALMUHAIDEB[3], AND KHALID ALISSA[3]

[1]College of Signals, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
[2]Saudi Aramco Cybersecurity Chair, Department of Computer Information Systems, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam 31441, Saudi Arabia
[3]Saudi Aramco Cybersecurity Chair, Department of Networks and Communications, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Dammam 31441, Saudi Arabia

Corresponding author: Abdullah M. Almuhaideb (amalmuhaideb@iau.edu.sa)

This work was supported by Saudi Aramco Cybersecurity Chair, Imam Abdulrahman Bin Faisal University.

ABSTRACT Advanced Persistent Threats commonly use Domain Generation Algorithms to evade advanced detection methods to establish communication with their command and control servers. To overcome the DNS protocol legitimacy breach, several Domain Generation Algorithms detection methods have been proposed. To solve the problem of DGA malware detection amicably, Deep Learning based detection schemes attracted researchers' interest recently. However, Deep Learning has already achieved optimal results and the gap is identified as fine-tuning of Deep Learning model hyper-parameters. The proposed solution is focusing on a model specific hyper-parameter known as the gradient optimizer. Gradient Optimisers are broadly categorised into Stochastic Gradient and Adaptive Moment based Gradient. Moment-based Gradient optimizer approaches are identified with suffering from weight decay and leading to poor generalization. Adaptive Moment (Adam) has improved with weight Decay as AdamW. To optimise moment-based gradient optimizers, Adam and AdamW are analyzed deeply. To optimize the functioning of AdamW, we present AdamW+, a novel solution for detecting DGA algorithms through re-implementing and nullifying the weight decay in AdamW. AdamW+ has been successfully implemented and shown promising results compared to Adam and AdamW optimizers in practice. AdamW+ preserved the properties of Adaptive Optimizer Adam while simplifying the weight decay implementation of AdamW. Empirical analysis has proved that AdamW+ has outperformed Adam and AdamW. The experimental result have substantiated that the proposed algorithm achieves the best accuracy result.

INDEX TERMS Deep learning optimizer, LSTM, Adam, AdamW, AdamW+.

I. INTRODUCTION

Nowdays, Advanced Persistent Threat (APT) is one of the most complex and hybrid cyber threats designed to target adversaries. APT is not a hostile attack in singularity but rather a successive stealthy cyber operation. Strategic level APT [1] targets nation-states' economies and Critical Information Infrastructures (CIIs) to steal national or corporate trade secrets. APT phases of cyber-attacks are sorted as sequential phases of attacks and are assorted as Cyber Kill Chain (CKC) [2]. CKC phase of Command and Control in which malicious actors exploit legitimate DNS protocol to evade any detection exercising malicious commands remotely. DNS protocol is an application layer protocol that uses both TCP and UDP for zone transfer and for name against associated IP information respectively. DNS protocol is abused by generating malicious domains in the garb of non-existent domains (NXDomains) [3]. One of such

The associate editor coordinating the review of this manuscript and approving it for publication was Yiming Tang.

malicious domains are then connecting the infected systems with pre-configured CC servers. The target system is infected by an explicitly designed malware to exploit the inherited trust of DNS protocol called Domain Generating Algorithm (DGA) [4]. DGA generates the bulk of NXDomain traffic along with the hidden malicious domains for connectivity with prefigured CC servers.

The earliest methods of DGA malware detection include blacklisting of such domains from bulk-generated domains by DGA communities [5] and detection of non-existent domain traffic using sequential hypothesis [6]. DGA detection has been advanced with Machine Learning and Deep Learning based solutions. More recently, DGA detection has been elevated using Deep Learning models like Long-short Term Memory (LSTM) and Convolution Neural Networks (CNN) models successfully. These Deep Learning models have brought considerable improvement in detection performance. As compared to LSTM, CNN and even their hybrid approaches have been applied and have outperformed all previous methods. The optimal performance of these DL models was based on solving the text classification problems (separating malicious domains from legitimate domains). Presently, the implementation of Deep Learning models with LSTM and CNN models has been applied profoundly for DGA Detection.

The Deep Learning models are dependent on various model parameters to classify and detect DGA-generated domains/ NXdomains. One such parameter us gradient optimizer which is an optimization algorithm and is considered the primer of the presented research work. This research is to observe and evaluate the efficient gradient optimizer algorithms for the LSTM model. Gradient optimizers are divided mainly into Stochastic Gradient (SGD) and Adaptive Moment-based Optimizers. Adaptive Moment based optimizers refine the moving averages more smoothly than SGD. Here, we employ an Adaptive Moment based gradient optimizer for text classification problems like malicious domain detection generated by DGA. DL models are normally set with Adaptive Moment optimizer (Adam) [7] as the default optimizer. It optimizes the LR and fastens the convergence of the training model to a point of stability. Adam is improved further by AdamW (Adam with fixed weight Decay) [8] which deploys weight decay separately than L2 regularization.

Deployment of the weight decay coefficient is considered a meager impact on the overall implementation of the optimizer itself, this is the gap identified in AdamW. Thus, we first implement Adam and AdamW optimizer with respect to focus on weight decay and then pitched our proposed optimizer AdamW+. In this proposed research work, the implementation of weight decay is presented in AdamW [8]. The optimized variation of AdamW with respect to weight decay is named AdamW+. Empirical analysis of text classification with a Deep Learning model for DGA detection is chosen for comparative analysis of adaptive moment-based optimization algorithms. Text classification-based DGA detection is implemented for AdamW+ optimizer with a comparison against the default Adam and AdamW optimizers. AdamW+ has shown that AdamW implementation become more efficient and has outperformed both adaptive moment-based optimizers like Adam and AdamW.

This study is divided into six sections. Section II covers related work on DGA detection encompassing two subsections of Deep Learning-based DGA detection and evolution of Adaptive Gradient Optimization Algorithms respectively. Section III identified the gap and highlights the proposed methodology of the subject research work. Section IV explains proposed methodology and its empirical implementation. Section V discusses the results and section VI discusses conclusive remarks with future directions.

II. RELATED WORK

A Domain Name may have maximal length of 253 characters as per RFC 1035 [9]. The length of the domain name varies with respect to DGA families and APT groups as depicted in Table 1. Moreover, detected DGA lengths are varying from 7 to 32 alphanumric characters using different generation schemes. The DNS request ''ffqrgedkmxbwb[.]ru'' from Table 1 is generated by a DGA embedded in Conficker malware.

TABLE 1. Comparative list of various malicious domains generated by different DGA families.

Samples/No of Characters	Associated DGA Family	Type of Generation Schemes
gdgdhdddjkdgh.com	Cryptolocker [10]	Arithmetic
hpbbydetwdqsscqtnvljufaau.com	Gameover [11] / P2P	Arithmetic
ffqrgedkmxbwb.ru/13	Conficker	Arithmetic
miodndu.ms	Necurs [12]	Arithmetic
sizyvob.com	simda [13]	Arithmatic
seekhecsfam.com	qakbot [14]	Arithmetic
nxnucfb.info	shifu [15]	Arithmetic
b83ed4877eec1997fcc39b7ae590007a.info	Bamital [16]	Hashing

As DGA Detection as it is backed by organised and professional APT groups, highly sophisticated Cyber criminals and Cyber state actors which is highly challenging. They collectively produce too much variants to keep evading advanced Cyber security solutions and products. Conficker was the most active malware family identified followed by Necurs [12] and Suppobox [17]. The Conficker malware was designed to infect and control millions of PCs worldwide by cyber criminals. It was structured with advanced program logics, peer-to-peer (P2P) coordination channel and domain generation algorithm (DGA) [18]. Figure 1 explains how a malicious Command and control server with an assumed IP of 111.222.111.222 is associated with one of the generated domains by DGA and connects the infected machine back with a listening malicious Command and control server. With such a spread in domain space manipulation, it became a significant challenge to detect DGA malware as it exploits

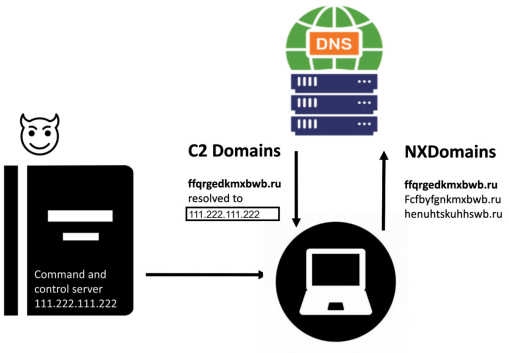


FIGURE 1. Malicious command and control server connecting with victim machine.

inherit weaknesses and loopholes in the Internet address and name space governance [19].

First Problem with domain names is its control whose geographical boundaries are transfrontier. This makes it a global menace with no centralized control due nonexistence of uniform regulations across the world and regions for domain registrations. TLDs reveal information to determine the identity of web entities the generated domains are registered. Despite this government are unable to take control over these domain registrars and have failed in implementing a coherent regulatory control. US Federal Agency FBI failed in one of their famous Cyber Operation Tovar to take over Russian domain registrars. Second problem is detection of minute ratio of DGA traffic compared to enormous legitimate DNS traffic. Effective DGA detection is becoming tough business to identify the prevailing rogue connections to malicious C&C servers. It asserts to disable security product as well as stops Windows from security patches and updates. This disables security update notifications and kills processes whose names match a list of 23 security products and security diagnosis tools.

As DGAs are associated with different APT groups to establish their CC servers covertly and abuse the DNS protocol legitimacy. Presently, DGA detection has evolved to be solved by most advanced Deep Learning models or Deep Neural Networks (DNNs). DGA malware authors are constantly evolving from random alphanumeric characters to word-lists or a word dictionary to make it more legitimate domain name to avoid detection by advance security solutions. Regardless of length or word-lists adopted by malware authors, DL models detects successfully the repetitions of specific characters or wordlists and associate its statistics with a DGA family/ group if it is generated by DGA.

A. DGA DETECTION WITH DEEP LEARNING

DGA malware produces varying alphanumeric patterns and character frequencies as well as wordlist based [20] domain names. Both patterns and character frequencies are cleverly designed by malware engineers to avoid detection and bypass advanced detection systems. Further, these patterns, frequencies, and word list-based domain names associate themselves with respective DGA families. The bulk volumes of malicious domain names generated by DGA malware confirm it as a potential candidate for Deep Learning models. Deep Learning models achieved better performance in DGA classification and detection due to having an inherent auto features extraction and better results over DGA detection. These DNN models generally consist of LSTM, CNN, and thier hybrid approaches [21]. The recent addition of Attention models with LSTM has further improved the detection performance [22]. A brief overview of research work based on these Deep Learning models is depicted in Table 2.

TABLE 2. Overview of DGA detection with individual deep learning models.

Advanced DGA Detection DL Techniques	Years	Research Work done
LSTM	2016	J. Woodbridge et al. [23]
	2018	R Vinayakumar et al. [24]
	2023	Hu, Xiaoyan et al. [25]
	2024	Tapsoba et al. [26]
CNN	2018	Duc tran et al. [27]
	2017	Joshua et al. [28]
	2018	W. Bush et al. [29]
LSTM with Attention / Hybrid approach	2019	Shaofanf Zhao et al. [30]
	2019	Y. Qiao et al. [31]
	2021	J. Namgung et al. [21]
	2024	BR S et al. [32]

Further, it is pertinent to mention that the Transformer model [22] undoubtedly are used in processing all input data at once to achieve long-range dependencies while LSTM model process input data sequentially and achieves both long-range (not upto scale of Transformer) and short-range dependencies. Further Transformers are used ideally in dealing with large corpus of data sets more likely for text translation and text summaries. As DGA Detection uses only domain names per sample with maximal length upto 100 characters as DGA malware authors want to mix their malicious traffic into legitimate network traffic and avoid overwhelming lengthy domain names. It is considered that domain name samples are much smaller comparatively to larger corpus datasets. Therefore, it is ideal to adopt LSTM models for malicious domain detection. However, a layer of Attention has been added in proposed LSTM model which has drawn connections between all input data and achieve same performance results as of Transformer used in larger corpus.

Deep Learning models learn to differentiate between legitimate and malicious domains using training dataset of both legitimate and malicious domain samples. Deep Learning models are fed with labeled samples of both legitimate domains and malicious domains for training and learning. In Deep Learning, LSTM models are considered ideal for text classification problems due to the inherent ability of memory correlations for past inputs. LSTM models are further augmented with Attention [31] have further improved the performance of DGA detection. Narrowing the spectrum of this research, LSTM with Attention models are selected to assess and optimize the performance of these models. In model parameters, the gradient optimizers are

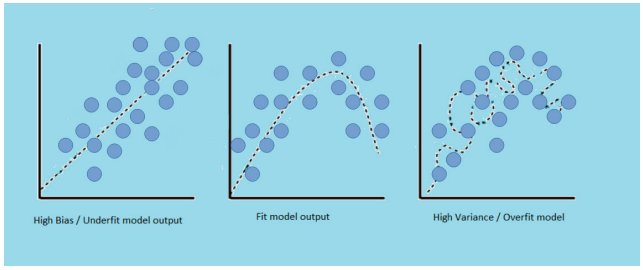


FIGURE 2. Models underfit, fit and overfit presentation [33].

selected as the core model parameter to be focused on and evaluated in light of available best-performing optimizer functions.

B. EVOLUTION OF ADAPTIVE GRADIENT OPTIMIZATION ALGORITHMS

Deep Learning models are ascertained as either the model is fit, under-fit, or over-fit during its training. Model generalization is observed with the convergence of Learning Rate (LR) culminating towards a point of stability. Deep Learning Model parameters with higher dimensions lead to higher non-linearity which supports a faster Learning Rate during the training. However, with higher parameter dimensions, a higher bias may also lead the model to become an under-fit model and a higher variance may lead to over-fitting of the model. To keep the bias and variance within limits and to obtain a fit Deep Learning model, the gradient is moved in direction of desired global minima (a minimal loss point) smoothly. Moving down the error slope or gradient of the Deep Learning model called Gradient Descent (GD), the learning rate determines the size of the step to reach the desired global minima. Learning Rate follows along the direction of the slope by a function descending down to reach the global minima. The role of the GD is to smooth the step-size of movement towards global minima. Framing this Deep Learning representational function as a stochastic function f and its parameters as θ, function f(θ) is optimized with the simplest approach as the Stochastic Gradient Descent (SGD) [34] is mapped as,

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla f(\theta) \tag{1}$$

where η is the LR which defines the required step size to reach the local minima and ∇f(θ)is the rate of change of parameters θ with respect to objective function f. Stochastic Gradient Descent (SGD) with moment (SGDM) added a fraction β to update the parameters' first moment as m. Upgrading SGD equation 1 to SGD with the first moment as m_t at time step t,

$$\theta_{t+1} = \theta_t - m_t \tag{2}$$

where,

$$m_t = \beta m_{t-1} + \eta \cdot \nabla f(\theta) \tag{3}$$

SGD with momentum is upgraded by the unveiling of Adagrad (adaptive gradient method) [35] which makes the gradient flexible to adapt the lower or higher Learning Rate (step sizes) instead of fixed step size with SGD. Adagrad has two main advantages, first, it is well suited for the sparsity of data and second, it adjusts the tuning of Learning Rate (step sizes) faster and eliminates manual tuning. Adagrad has perceived the concept of adaptive Learning Rate from the concept of moving averages. Adagrad is presented mathematically in equation 4 as,

$$\theta_{t+1,i} = \theta_t - \frac{\eta}{\sqrt{G_{t,i} + \epsilon}} \cdot gt \tag{4}$$

$G_{t,i}$ is the sum of squares of gradients g_t at time t and i wrt parameters θ_t. The equation has clearly depicted how the Learning Rate (step size) is now controlled by the square root of gradients in action and ϵ is a very small number to avoid division by zero. AdaDelta [36] and RMSprop [37] (which are almost identical) have introduced fixed weight size accumulation, further improving with a sum of squared gradients which is the decaying average of all past squared gradients, it actually introduced second order moment estimation as v_t after first order m_t as,

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} \cdot gt \tag{5}$$

Adaptive Moment Estimation (Adam) [7] algorithm has further clarified and improved the adaptive Learning Rate by computing the decaying averages of past gradients and past squared gradients as m_t and v_t respectively as,

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\eta \cdot \nabla f(\theta) \tag{6}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\eta \cdot \nabla f(\theta) \tag{7}$$

and essentially the bias correction as $\hat{v}_t$ and $\hat{m}_t$ to avoid the output being influenced by zero initialization.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \cdot \hat{m}_t \tag{8}$$

However, both SGD with momentum and Adam Learning Rate are observed to be generalizing poorly over a diverse set of deep learning models due to some inherent problems of adaptive gradient methods. In the presented case, weight decay is identified as the propelling factor of these problems and its implementation is considered undermined. The same is fixed in Adam from equation 8 as AdamW in [8] and represented as,

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \cdot \hat{m}_t + w_t \theta_{t-1} \tag{9}$$

The prime motive of equation 9 has been identified as the research gap where weight decays w_t needs decoupling from L2 regularization [8] and re-implementation as a stand-alone parameter in Adam. The Learning Rate is an adaptive parameter while weight decays w_t works as a coefficient (a small numerical value).

III. PROPOSED METHODOLOGY - WEIGHT DECAY SIMPLIFICATION

Weight decay is used to regularize the DL models and is multiplied with model weights with a small numerical fraction during updating new weights. Weight decay was considered an integral part of L2 regularization which was justified in [8] by decoupling it from L2 regularization and specifically implementing equation 9. Analyzing the decoupling of weight decay w_t deeper, it is learned that equation 9 may be further simplified as;

$$\theta_t = \theta_{t-1}(1 - w_t) - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t \tag{10}$$

As weight decay parameter in equation 10 is just a numerical figure and is applied in fractions of logarithmic values such as 0.1, 0.01, 0.001, and so on. These values for instance if added in equation 11, will be added as coefficients of 0.9, 0.99, and 0.999, and so on. This will result in parameter θ_t in a meagre correction as the case of weight decay w_t is generally started implementing from 0.001, 0.0001, and so on. Continuing on equation 10, if we apply w_t as 0 equating the meagre value to null, theoretically we regain Adam as a result of neutralizing the parameter w_t to zero. However, rather than using Adam again, we implemented AdamW with w_t equal to 0 in equation 10. This led us to discover that $w_t = 0$ is a more potent implementation of AdamW and the same reimplementation is named AdamW+. After identifying the novel optimization approach, the three optimizers Adam, AdamW, and AdamW+ are tested in solving the text-based classification problem of DGA Detection. The three optimizers have been implemented on LSTM with Attention to DGA detection and subsequently comparing and evaluating the outcomes of the 3 optimizers.

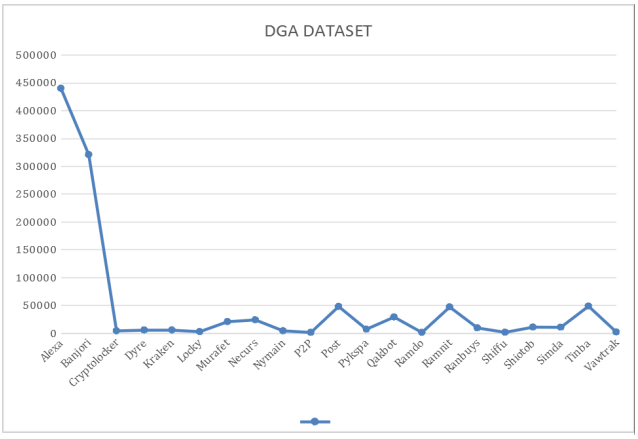


FIGURE 3. Dataset visual breakdown.

A. ADAMW+ PSEUDO CODE

Algorithm 1 explains the simplified implementation as keeping the implementation of AdamW with nullifying the weight decay coefficient. Steps 1 and 2 define the optimizer parameters and step 3 defines time variables. Step 4 highlights the AdamW optimizer with the first and second moment including weight decay implementation. Step 5 factorized the weight decay from AdamW and nullified its coefficient. The same is implemented in the code at [38].

Algorithm 1 Weight Decay Plus AdamW+, off Shooting of AdamW

1: $StochasticFunction = f, parameters\theta \leftarrow f(\theta)$
2: $LearningRate = \eta \leftarrow step - size$
3: $Rateofchangeoff(\theta) = \nabla f(\theta)$
4: $InitializingTimeStep, i = 0 \leftarrow i = t$
5: $First - Order - Moment = m_t$
6: $Second - Order - Moment = v_t$
7: $Weight - Decay = w_t$
8: AdamW:

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t + w_t\theta_{t-1} \tag{11}$$

9: AdamW factorized w_t:

$$\theta_t = \theta_{t-1}(1 - w_t) - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t \tag{12}$$

10: Nullifying w_t in above equation $\leftarrow$ AdamW+:

$$\theta_t = \theta_{t-1} - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t \tag{13}$$

IV. EMPIRICAL SETUP

LSTM with Attention model is considered one of the advanced approaches in solving text classification problems and the same is adopted for DGA detection problems. The experimental setup started with legitimate domain samples from Alexa [39] and 20 x DGA families samples from Bamabanek [40]. All models are implemented with a dataset split of 75% training and 25% testing samples. All the datasets, code repositories, and results are available at [38] for reference and future work. Training and testing datasets are composed of one Legitimate domain dataset against 20 classes of varying DGA families. The total dataset samples are 1.435 million samples which consist of 0.6 million from Alexa's top million domain names as legitimate domain names and the rest 0.83 million malicious domain samples composed of twenty DGA families. A visual breakdown of the dataset is projected in Figure 3. Implementation details are available and accessible at [41] for future referencing and research work ups. Implementation details include code details, used legitimate and malicious DGA datasets and obtained results. As LSTM is a state and context-aware neural network, it is proficient in the detection of temporal associations between texts. LSTM obtains contextual vectors of input sequences. The attention mechanism with the LSTM model further improves the longer dependencies. DGA samples are passed through the Seq2Seq encoder which compresses input to a fixed length of a context vector. Each model as depicted accumulates the score of given input samples to classify it

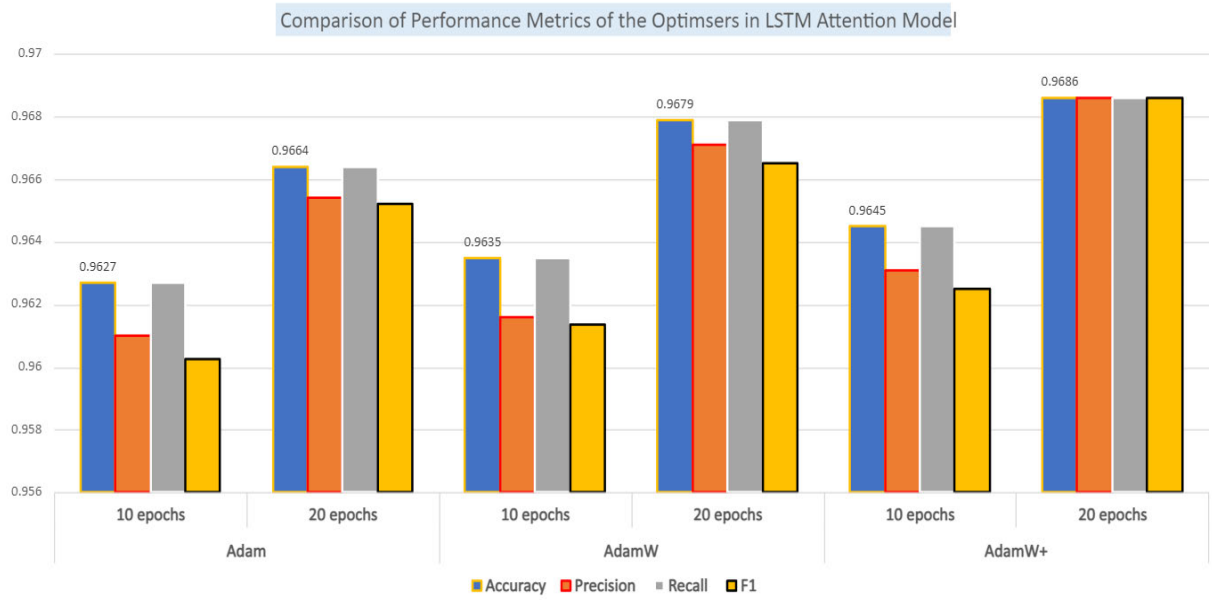


FIGURE 4. LSTM attention model performance metrics comparison for 3 optimisers.

either as a legitimate or malicious domain. As LSTM retains the stateful property, however still faces information loss in case of longer sequences or dependencies. This information loss is addressed with the addition of the Attention layer. LSTM output is further fine-tuned with the Attention layer. At the last layer, this binary classification of legitimate and DGA domains is further classified using the Softmax function to a specific DGA family. For multi-class datasets, we use a multi-classification model at the final output layer. All the output of the LSTM model is processed at the Fully Connected layer with Softmax giving the output score of each class. Softmax output gives the alignment score of various outputs and classifies them into different classes based on the closeness of these defined alignment scores.

V. RESULTS AND DISCUSSION

A. RESULTS

Performance metrics of these Deep Learning models are measured for DGA Detection with the adoption and implementation of Adam, AdamW, and AdamW+ optimizers. Two iterations of 10 epochs and 20 epochs are being run to obtain and compare the achieved results respectively.

Table 3 and Table 4 are showing an overall picture of performance metrics and computational proficiency of each model with deeper performance visibility. Generally, proposed optimizer AdamW+ has shown optimal performance in the detection of all 20 DGA families against Alexa domain names. A broader overview of all performance metrics outcomes of each model which are depicted in Table 3 and 4, their graphical presentations are projected in Figure-4 respectively and collectively. A closer observation graphical projection of performance metrics in Figure-5 shows that proposed adaptive optimiser AdamW+ has outperformed than legacy optimisers Adam and AdamW. The same can be further validated from a convergence of performance metrics in Figure-5 (c) which is substantiating the performance, by achieving 97% which is more smooth and more stable than Figure-5 (a & b).

TABLE 3. LSTM attention model performance comparison of 3 optimiser with 10 epochs.

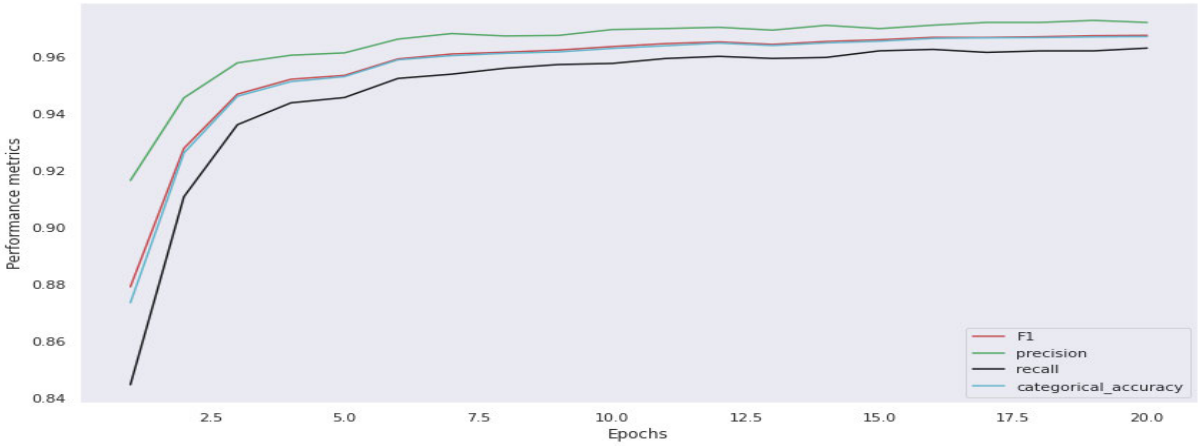
LSTM Attention Model	Adam (10Epochs)	AdamW (10Epochs)	AdamW+ (10Epochs)
Accuracy	0.9627	0.9635	0.9645
Precision	0.9610	0.9616	0.9631
Recall	0.9627	0.9635	0.9645
F1 Score	0.9602	0.9614	0.9625

TABLE 4. LSTM attention model performance comparison of 3 optimiser with 20 epochs.

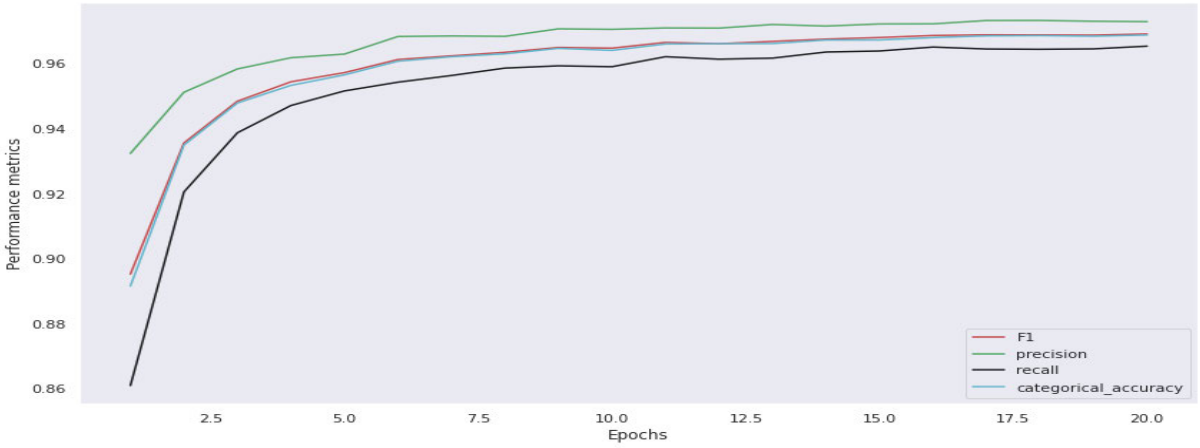
LSTM Attention Model	Adam (20Epochs)	AdamW (20Epochs)	AdamW+ (20Epochs)
Accuracy	0.9664	0.9679	0.9686
Precision	0.9654	0.9671	0.9674
Recall	0.9664	0.9679	0.9686
F1 Score	0.9652	0.9665	0.9673

Training and validation accuracy as well as training and validation loss of each model are projected in Figure 6 (a, b & c) to identify how well the model is fit. It is evident that the accuracy and loss curves of all the depicted models have converged at an optimum value.

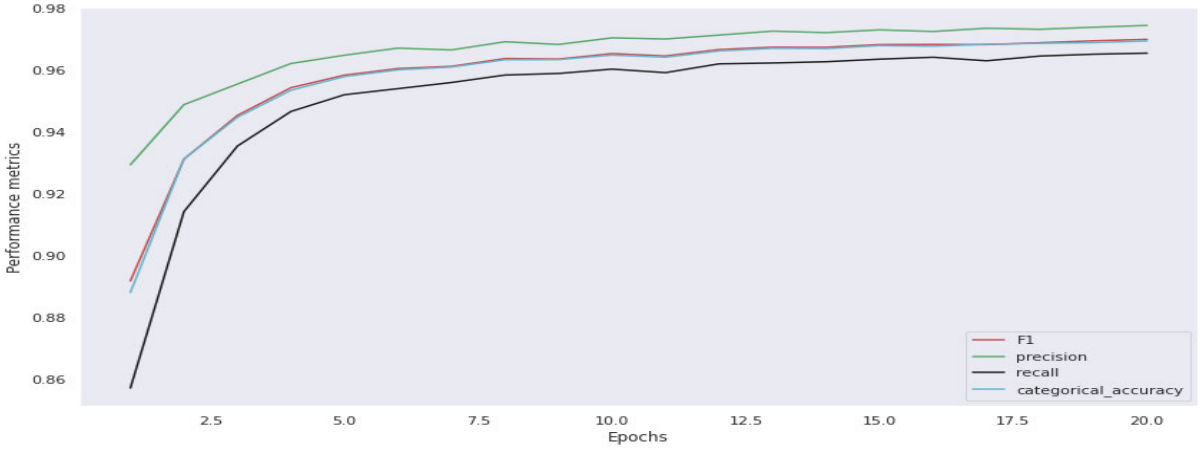
As two iterations of the 3 models have been run, Figure 7 (a, b & c) has shown performance metrics of Precision, Recall, and F1 score for 10 and 20 epochs respectively for the three selected optimizers.



(a) LSTM Attention with Adam-20 Epochs



(b) LSTM Attention with AdamW-20 Epochs



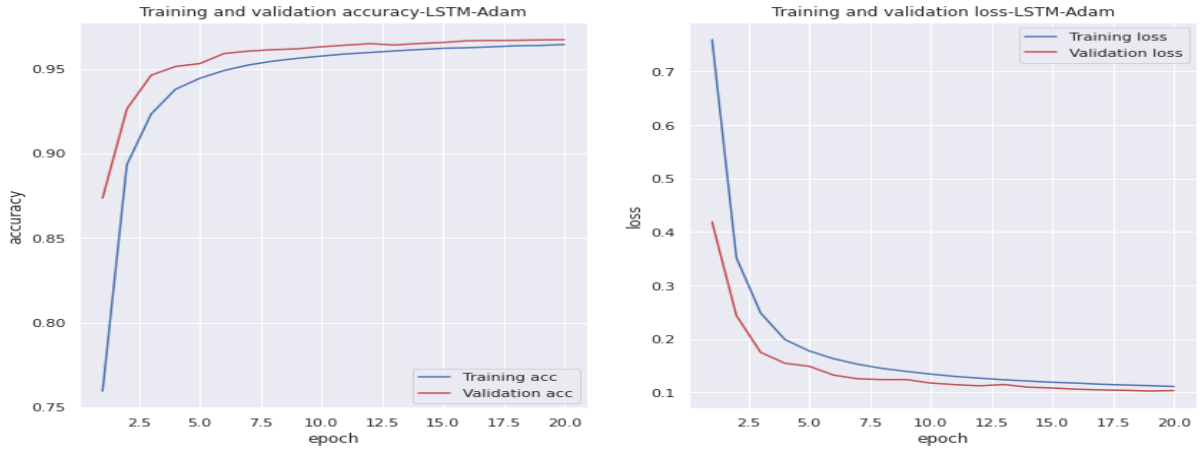
(c) LSTM Attention with AdamW+-20 Epochs

FIGURE 5. **Performance metrics comparison.**

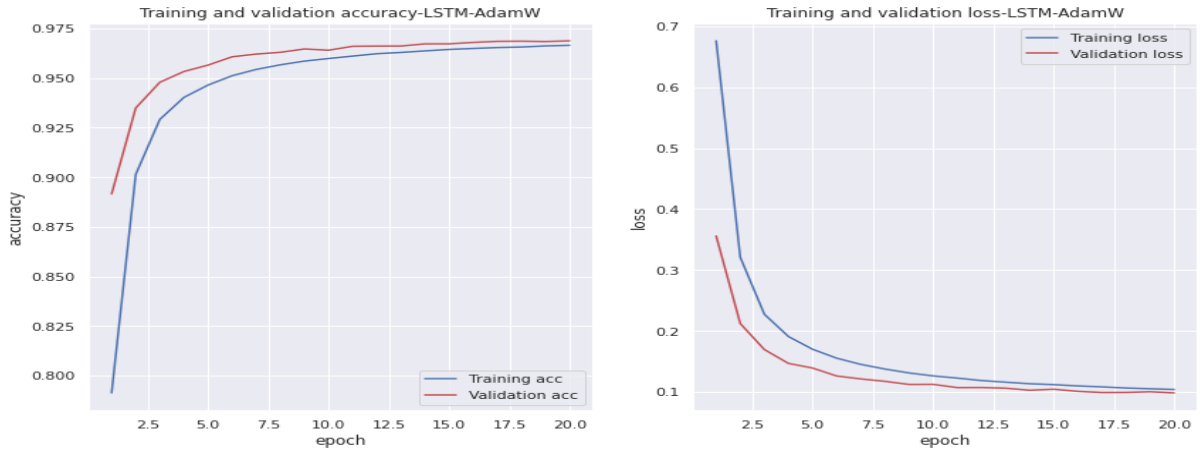
The correlation matrix of the LSTM attention model for proposed AdamW+ optimizers executed on 10 and 20 epochs are shown in Fig.8, and Fig.9, respectively.

B. DISCUSSION

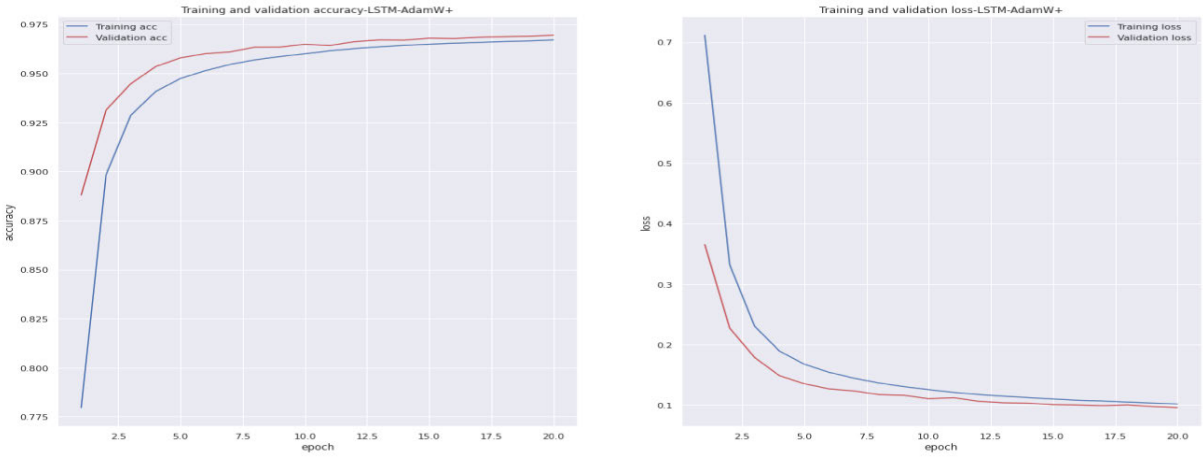
Analyzing all results in presented tables and figures reveal that DGA detection using LSTM with Attention model have achieved optimal performance. Further, Optimizers



(a) LSTM Attention with Adam



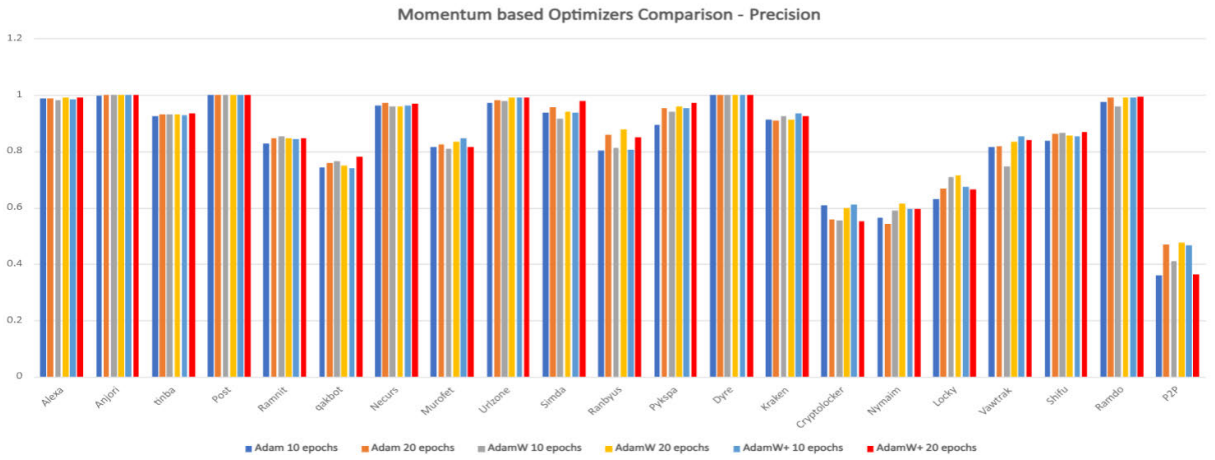
(b) LSTM Attention with AdamW



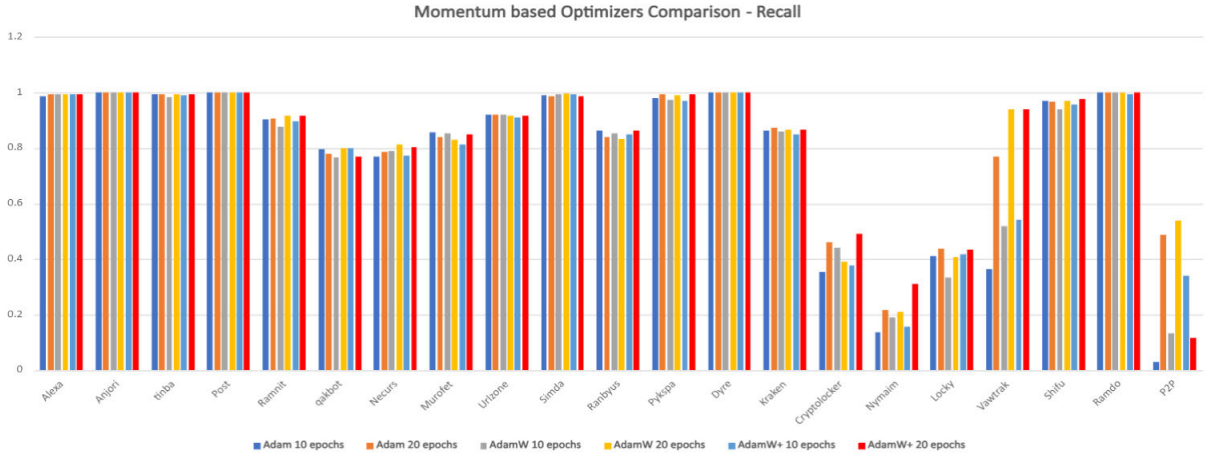
(c) LSTM Attention with AdamW+

FIGURE 6. **Training and validation accuracy vs Training and validation accuracy loss comparison of 3 optimizers.**

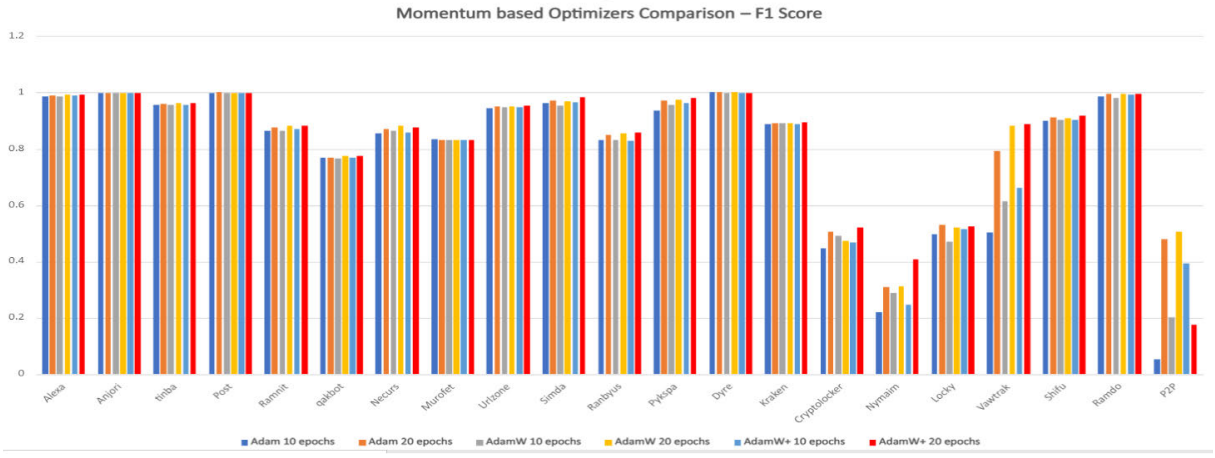
iteratively decrease the loss function to modify the weights of given gradients. Efficient optimizers force a model to generalize faster and the choice of optimizer influences the performance of the model. Comparing Adam, AdamW and AdamW+ optimizers have achieved significant results and shown to be close competitors. However, the default optimizers are outperformed by the proposed AdamW+ with optimal performance. Overall, LSTM with Attention model deployed with AdamW+ results has shown significant progress in all performance metrics in DGA detection. It has



(a)



(b)



(c) c

FIGURE 7. **Performance comparison of (a) Precision (b) Recall (c) F1 score for 10 and 20 epochs.**

been proved that weight decay or regularization term does not end up in the moving averages and is thus only a meagre proportion to the weight itself. Therefore, an improved version of Adam called AdamW+ is simulated where the weight decay is negated in the parameter-wise step size of AdamW. Fine-tuned adaptive AdamW+ optimizer is faster as compared to Adam and AdamW in terms of generalization performance. Moreover, it is also assessed that Adam and AdamW being good competitors have not underperformed but have been identified with a gap of improvement in

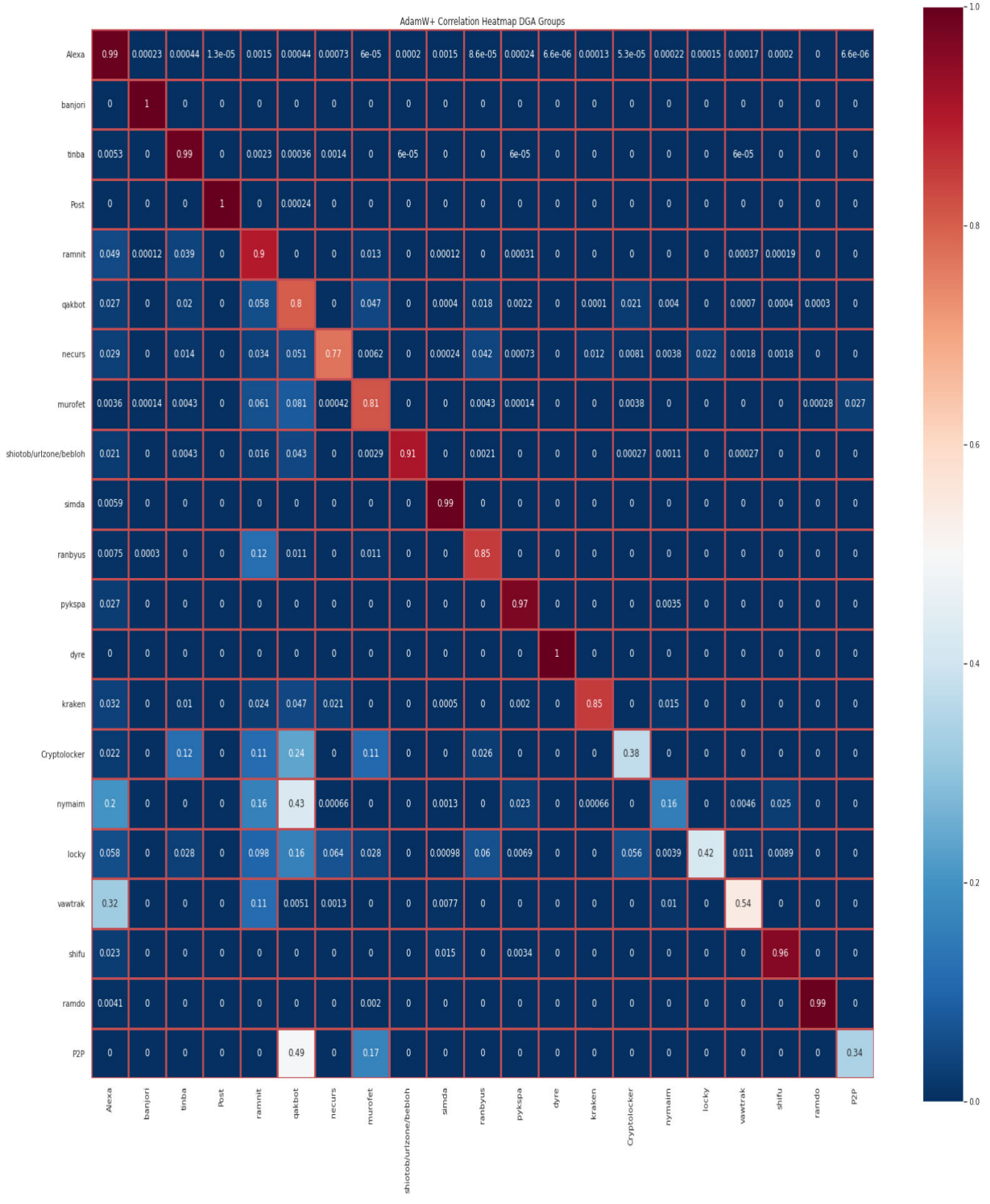


FIGURE 8. Correlation matrix LSTM attention model with AdamW+ - 10 epochs.

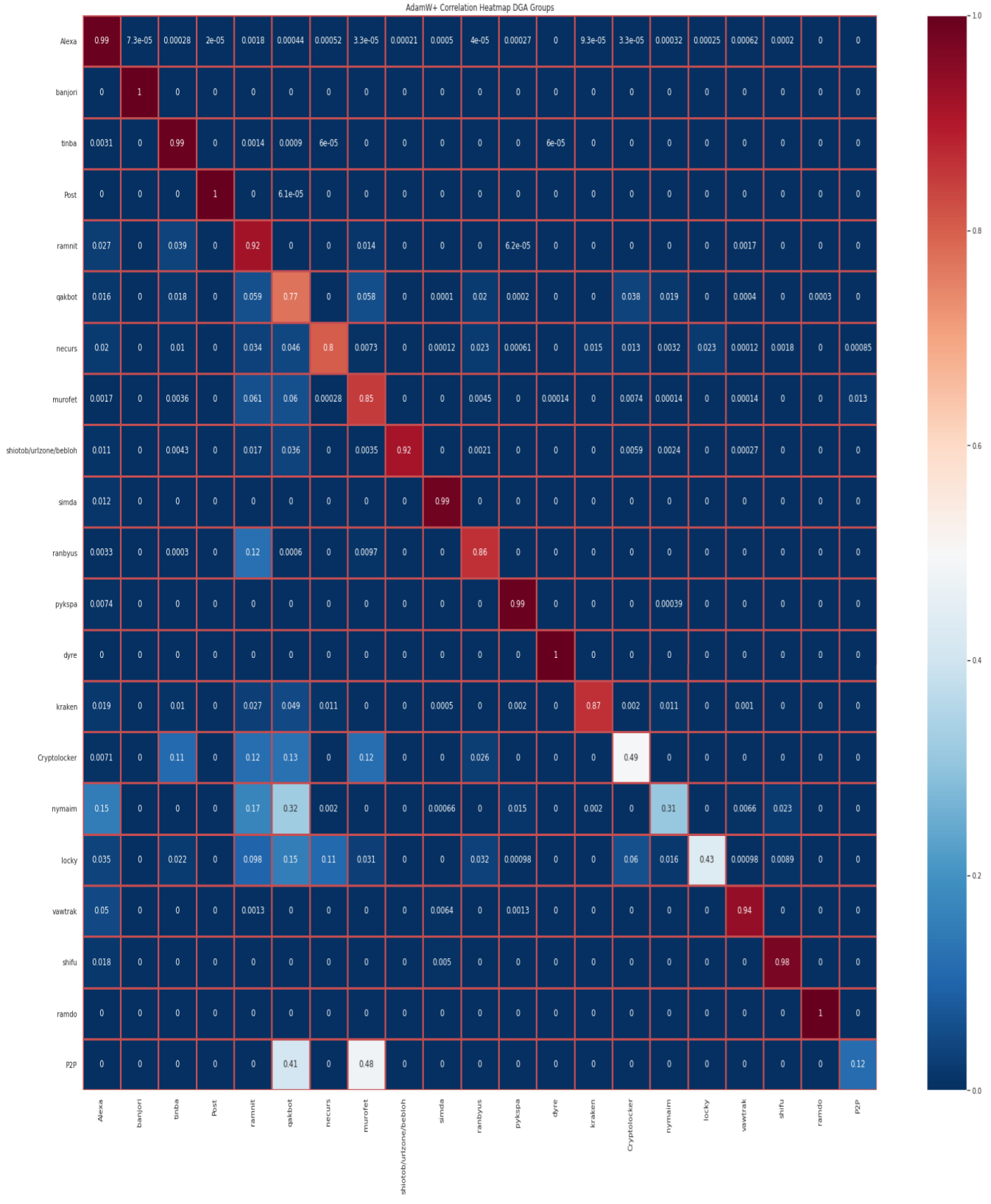


FIGURE 9. Correlation matrix LSTM attention model with AdamW+ – 20 epochs.

fine-tuning the AdamW optimizer. The proposed AdamW+ has preserved the base properties of adaptive optimizer Adam by simplifying AdamW back to Adam step size for updating weight vectors, resulting in an improved generalization of the model.

C. FUTURE WORK

The subject research broadly covers cyber security paradigm and is manifold. It focuses on DGA detection from traditional methods to advanced deep learning algorithms. In deep learning algorithms LSTM has been selected to focus on a deep learning parameter like gradient optimizer. Adaptive Moment Estimator (Adam) optimizer has been introduced against Stochastic Gradient Descent (SGD) in 2014. Adam has been upgraded with Adam with Wieght Decay in 2018. The concept of wieght decay is accepted, however core of research is that weight decay is too small and did not make any difference. AdamW+ is discarding the weight decay in AdamW optimizers. It is completely a new dimension and lacks cross researcher's data. It is therefore considered prudent that future work of applying AdamW+ in other deep learning models like CNN etc may be substantiated with this presented research work as both of its cross-research data and presented results.

VI. CONCLUSION

This paper presented regulation techniques to improve model performance in the areas of DGA detection. The proposed approach was compared and analyzed with the core performance metrics. Through implementation and experimental result, we can demonstrate that a new dimension/approach in optimizer has been gained as well as these optimizers have shown optimal performance under both prototype datasets and real-world problems. Future works may include switching the proposed optimizer AdamW+ in larger text classification problems as well as other DL models like CNN and GANs. The same approach may also be compared with Stochastic Gradient Descent (SGD) as SGD has shown better performance in some approaches.

REFERENCES

[1] A. Ahmad, J. Webb, K. C. Desouza, and J. Boorman, "Strategically-motivated advanced persistent threat: Definition, process, tactics and a disinformation model of counterattack," *Comput. Secur.*, vol. 86, pp. 402–418, Sep. 2019.

[2] A. Dorian Wong, "Detecting domain-generation algorithm (DGA) based fully-qualified domain names (FQDNs) with Shannon entropy," 2023, *arXiv:2304.07943*.

[3] J. Ahmed, H. H. Gharakheili, and V. Sivaraman, "Learning-based detection of malicious hosts by analyzing non-existent DNS responses," in *Proc. GLOBECOM IEEE Global Commun. Conf.*, Dec. 2022, pp. 3411–3416.

[4] Q. Abu Al-Haija, M. Alohaly, and A. Odeh, "A lightweight double-stage scheme to identify malicious DNS over https traffic using a hybrid learning approach," *Sensors*, vol. 23, no. 7, p. 3489, 2023.

[5] M. Kührer, C. Rossow, and T. Holz, "Paint it black: Evaluating the effectiveness of malware blacklists," in *Proc. Int. Workshop Recent Adv. Intrusion Detection*. Germany: Springer, 2014, pp. 1–21.

[6] S. Krishnan, T. Taylor, F. Monrose, and J. McHugh, "Crossing the threshold: Detecting network malfeasance via sequential hypothesis testing," in *Proc. 43rd Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2013, pp. 1–12.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[8] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in Adam," in *Proc. ICLR*, 2018, pp. 1–14.

[9] *Domain Names—Implementation and Specification*, document RFC 1035, RFC Editor, Jul. 1995. [Online]. Available: https://www.ietf.org/rfc/rfc1035.txt

[10] M. M. Khan, M. F. Hyder, S. M. Khan, J. Arshad, and M. M. Khan, "Ransomware prevention using moving target defense based approach," *Concurrency Comput., Pract. Exper.*, vol. 35, no. 7, p. e7592, 2023.

[11] M. A. Kazi, S. Woodhead, and D. Gan, "An investigation to detect banking malware network communication traffic using machine learning techniques," *J. Cybersecurity Privacy*, vol. 3, no. 1, pp. 1–23, 2023.

[12] R. Bapat, A. Mandya, X. Liu, B. Abraham, D. E. Brown, H. Kang, and M. Veeraraghavan, "Identifying malicious botnet traffic using logistic regression," in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2018, pp. 266–271.

[13] V. S. Priyamvada Davuluru, B. Narayanan Narayanan, and E. J. Balster, "Convolutional neural networks as classification tools and feature extractors for distinguishing malware programs," in *Proc. IEEE Nat. Aerosp. Electron. Conf. (NAECON)*, Jul. 2019, pp. 273–278.

[14] E. Brumaghin, M. Graziano, and N. Mavis, "Squirrelwaffle leverages Malspam to deliver Qakbot, cobalt strike," *Talos Blog.*, vol. 28, p. 2022, Feb. 2021.

[15] X. Shen, X. Zhang, and Y. Chen, "Deep learning powered adversarial sample attack approach for security detection of DGA domain name in cyber physical systems," *IEEE Wireless Commun.*, vol. 29, no. 2, pp. 16–21, Apr. 2022.

[16] W. Fang, "Real time botnet detection system based on machine learning algorithms," in *Proc. 2nd Conf. High Perform. Comput. Commun. Eng. (HPCCE)*, vol. 12605, 2023, pp. 226–234.

[17] D. Plohmann, K. Yakdan, M. Klatt, J. Bader, and E. Gerhards-Padilla, "A comprehensive measurement study of domain generating malware," in *Proc. USENIX Secur. Symp.*, vol. 10, 2016, pp. 1–17.

[18] G. Alley-Young, "Conficker worm," in *The Handbook of Homeland Security*. Boca Raton, FL, USA: CRC Press, 2023, pp. 173–179.

[19] *Internet Governance*. Accessed: Jan. 15, 2024. [Online]. Available: https://www.itu.int

[20] R. A. R. Mahmood, A. Abdullah, M. Hussin, and N. I. Udzir, "Dictionary-based DGAs variants detection," in *Advances on Intelligent Informatics and Computing* (Lecture Notes on Data Engineering and Communications Technologies), vol. 127, F. Saeed, F. Mohammed, and F. Ghaleb, Eds. Cham, Switzerland: Springer, 2022, doi: 10.1007/978-3-030-98741-1_22.

[21] J. Namgung, S. Son, and Y.-S. Moon, "Efficient deep learning models for DGA domain detection," *Secur. Commun. Netw.*, vol. 2021, Jan. 2021, Art. no. 8887881.

[22] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, and M. Funtowicz, "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Syst. Demonstrations*, 2020, pp. 38–45.

[23] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Predicting domain generation algorithms with long short-term memory networks," 2016, *arXiv:1611.00791*.

[24] R. Vinayakumar, K. Soman, and P. Poornachandran, "Detecting malicious domain names using deep learning approaches at scale," *J. Intell. Fuzzy Syst.*, vol. 34, no. 3, pp. 1355–1367, 2018.

[25] X. Hu, H. Chen, M. Li, G. Cheng, R. Li, H. Wu, and Y. Yuan, "ReplaceDGA: BiLSTM based adversarial DGA with high anti-detection ability," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 4406–4421, 2023.

[26] A. R. Tapsoba, T. F. Ouédraogo, and W.-B. S. Zongo, "Analysis of plaintext features in DoH traffic for DGA domains detection," in *Proc. Int. Conf. Inf. Technol. Syst.* Germany: Springer, 2024, pp. 127–138.

[27] D. Tran, H. Mac, V. Tong, H. A. Tran, and L. G. Nguyen, "A LSTM based framework for handling multiclass imbalance in DGA botnet detection," *Neurocomputing*, vol. 275, pp. 2401–2413, Jan. 2018.

[28] J. Saxe and K. Berlin, "EXpose: A character-level convolutional neural network with embeddings for detecting malicious URLs, file paths and registry keys," 2017, *arXiv:1702.08568*.

[29] Z. Dai, C. Xiong, J. Callan, and Z. Liu, "Convolutional neural networks for soft-matching n-grams in ad-hoc search," in *Proc. 11th ACM Int. Conf. Web Search Data Mining*, Feb. 2018, pp. 126–134.

[30] S. Zhou, L. Lin, J. Yuan, F. Wang, Z. Ling, and J. Cui, "CNN-based DGA detection with high coverage," in *Proc. IEEE Int. Conf. Intell. Secur. Inform. (ISI)*, 2019, pp. 62–67.

[31] Y. Qiao, B. Zhang, W. Zhang, A. K. Sangaiah, and H. Wu, "DGA domain name classification method based on long short-term memory with attention mechanism," *Appl. Sci.*, vol. 9, no. 20, p. 4205, 2019.
[32] S. Harishkumar and R. S. Bhuvaneshwaran, "Enhanced DGA detection in BotNet traffic: Leveraging N-gram, topic modeling and attention BiLSTM," Res. Square, Version 1, Feb. 2024, doi: 10.21203/rs.3.rs-3981569/v1.
[33] *Underfitting Vs Just Right Vs Overfitting in Machine Learning*. Accessed: Jan. 15, 2024. [Online]. Available: https://www.kaggle.com/discussions/getting-started/166897
[34] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999.
[35] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 1–39, 2011.
[36] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*.
[37] T. Tieleman and G. Hinton, "Lecture 6.5-RMSPROP: Divide the gradient by a running average of its recent magnitude," *COURSERA, Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.
[38] *LSTM Optimizers*. Accessed: Jan. 15, 2024. [Online]. Available: https://www.kaggle.com/code/awaisjaved/lstm-model-with-optimisers/edit
[39] (May 1, 2022). *Alexa*. [Online]. Available: https://alexa.com
[40] *Bamabanek*. Accessed: Jan. 19, 2024. [Online]. Available: https://www.bambenekconsulting.com/
[41] *LSTM Optimizers*. Accessed: Jan. 19, 2024. [Online]. Available: https://github.com/awaiswill/Optimization_using_Stochastic_Optimization

AWAIS JAVED received the M.S. degree in wireless communication from Pakistan Naval Engineering College (PNEC), National University of Science and Technology (NUST), Karachi, in 2015. He is currently pursuing the Ph.D. degree in information security with NUST, Islamabad, Pakistan. He has published various publications in well reputed journals. His research interests include wireless communication, information security, and deep learning models, in particular wireless communication encompassing 5G, SDRs, packet radio, and domestic wireless networks. His information security interests include red teaming skills, advanced persistent threats (APTs), cyber kill chains (CKC), DGAs and MITRE's techniques, tactics and procedures (TTPs), SDNs security, and cloud computing security, and his deep learning models interests include LSTM models, CNN models, GANs models, and transformers.

IMRAN RASHID received the B.E. degree in electrical (telecom) engineering from the National University of Sciences and Technology, Pakistan, in 1999, the M.Sc. degree in telecomm engineering (optical communication) from DTU, Denmark, in 2004, and the Ph.D. degree in mobile communication from The University of Manchester, U.K., in 2011. He has been qualified for four EC-Council certifications, i.e., a Certified Ethical Hacker, a Computer Hacking Forensic Investigator, an EC-Council Certified Security Analyst, and an EC-Council Certified Incident Handler. He is also a Certified EC-Council Instructor and has conducted numerous trainings. His research interests include mobile and wireless communication, MIMO systems, compressed sensing for MIMO OFDM systems, massive MIMO systems, M2M for mobile systems, cognitive radio networks, cyber security, and information assurance.

SHAHZAIB TAHIR (Senior Member, IEEE) received the B.E. degree in software engineering from Bahria University, Islamabad, Pakistan, in 2013, the M.S. degree in information security from NUST, Islamabad, in 2015, and the Ph.D. degree in information engineering from the City, University of London, U.K., in 2019. He was a Research Fellow with the City, University of London. He is currently an Associate Professor with the Department of Information Security, NUST. His research interests include applied cryptography and cloud security. He has been a TPC member of many international IEEE conferences. He is an Alumni of Innovate U.K. CyberASAP. He is a Reviewer of IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, *IEEE Communications Magazine*, *Computers and Security* (Elsevier), IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS, IEEE ACCESS, IEEE ICC, *Future Generation Computer Systems* (Elsevier), *Cluster Computing* (Springer), *Sadhna* (Springer), and *Science China Information Sciences* (Springer).

SAQIB SAEED (Senior Member, IEEE) received the B.Sc. degree (Hons.) in computer science from International Islamic University Islamabad, Pakistan, in 2001, the M.Sc. degree in software technology from Stuttgart Technology University of Applied Sciences, Germany, in 2003, and the Ph.D. degree in information systems from the University of Siegen, Germany, in 2012. He is currently an Associate Professor with the Department of Computer Information Systems, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia. He is also a Certified Software Quality Engineer with American Society of Quality. His research interests include human-centered computing, data visualization and analytics, software engineering, information systems management, and digital business transformation. He is an Associate Editor of IEEE ACCESS and *International Journal of Public Administration in the Digital Age*, besides being a member of the advisory boards of several international journals.

ABDULLAH M. ALMUHAIDEB received the B.S. degree (Hons.) in computer information system from King Faisal University, Saudi Arabia, in 2003, and the M.S. (Hons.) and Ph.D. degrees in network security from Monash University, Melbourne, Australia, in 2007 and 2013, respectively. He is currently an Associate Professor of information security, the Supervisor of Saudi Aramco Cybersecurity Chair, and the Dean of the College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Saudi Arabia. He has published two patents and more than 40 scientific articles in journals and premier ACM/IEEE/Springer conferences. His research interests include mobile security, authentication and identification, and ubiquitous wireless access.

KHALID ALISSA received the Ph.D. degree in information security from QUT, Australia. He is currently an Assistant Professor of cyber security and digital forensics program with the College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University (IAU), where he is also the Dean of the Information and Communication Technology. He is an Information Security Consultant. His research interests include social engineering, access control, information security, and network and cloud technologies.