Performance Analysis of Deep Learning Models for DGA Domain Name Detection

RANJANA B NADAGOUDAR[1], DR. SUJATHA P TERDAL[2]
[1]*Assistant Professor, Visvesvaraya Technological University*
[2]*Professor, PDA College of Engineering*

Abstract-In recent years, the cyber security realm has experienced a notable uptick in reported incidents, particularly concerning malware attacks. These assaults, often orchestrated through botnets networks of compromised devices ranging from computers to IoT gadgets have become go-to tools for cybercriminals. Botnets facilitate a wide array of malicious activities, from DDoS assaults to spam dissemination, data breaches, click fraud, and identity theft. The proliferation of Domain Generation Algorithm (DGA) generated domain names poses a significant challenge in cyber security due to their role in evading traditional detection mechanisms. Traditional intrusion detection systems, reliant on signature-based approaches, find themselves struggling to keep up with the escalating sophistication of botnets. To address this challenge, the proposal suggests harnessing the power of machine learning and deep learning models. The deep learning models, such as Recurrent Neural Networks (RNN), Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) have shown promise in addressing this challenge. These advanced techniques hold promise in revolutionizing cyber security defense mechanisms, offering a proactive approach to combat the ever-evolving modern cyber threats. However, the demand persists for streamlined models capable of upholding high detection accuracy while conserving computational resources. In this study, we conduct a comprehensive performance evaluation of these models for DGA generated domain name classification and detection. Through extensive experimentation and analysis, we assess the accuracy, precision, recall, and computational efficiency of each model. Our findings provide valuable insights into the effectiveness of RNN, LSTM, and GRU models in achieving high detection accuracy with reduced computational overhead. The proposed model gated recurrent model has outperformed compared to all other deep learning models and achieves high accuracy and detection rate.

Index Terms- Deep learning, Cyber Security, Domain name generation (DGA), Recurrent Neural Networks (RNN), Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM).

I. INTRODUCTION

With the evolution of cyber attacks, particularly those utilizing Domain Generation Algorithms (DGAs), poses significant challenges to cyber security. The dynamic nature of DGAs makes it difficult for traditional static blacklists to effectively block malicious domains. Security organizations indeed face a daunting task in trying to keep up with the constantly evolving tactics of cyber attackers. A notable strategy utilized by malware and botnets involves the adoption of DGAs to establish their Communicate & Control (C & C) channels [4]. By leveraging DGAs, malicious actors aim to maintain flexibility and evade detection by traditional blacklists. DGAs operate by generating a vast array of domain names based on a predetermined seed shared between the attacker's master server and client devices. From this expansive pool, a select subset of domains is chosen by the attacker for registration. Client devices then systematically traverse this domain set, establishing connections with the registered domains upon query. Through this process, a C & C channel is established between the client and the master server, enabling communication and control over the compromised network.

The deep learning has emerged as a powerful tool across various domains, demonstrating remarkable performance in tasks ranging from image classification to speech recognition, sentiment analysis, and recommender systems [7]. In the realm of cyber security, the application of deep learning methods has proven particularly promising; especially in combating sophisticated threats like those posed by Domain Generation Algorithms (DGAs). By leveraging the capabilities of deep learning, cyber security Professionals can improve their capacity to detect and mitigate evolving threats, including those utilizing sophisticated techniques like DGAs [19]. The

integration of deep learning methods into cyber security frameworks holds significant promise for bolstering defence mechanisms and safeguarding against emerging cyber risks. The dynamic nature of the domains generated by DGAs presents a key challenge [5]. Attackers can readily alter registered domains, making it exceedingly difficult for defenders to pre-emptively block all potential sources of malicious activity using static blacklists. Consequently, conventional defensive measures often prove ineffective in thwarting DGA-based attacks. In response to the drawbacks of conventional methods, researchers have devised classifiers utilizing deep neural network architectures such as Recurrent Neural Networks (RNNs), Gated Recurrent Units (GRUs) and Long Short-Term Memory Networks (LSTMs) [12].

This paper explores the application of GRUs for DGA domain name classification and detection [13]. By leveraging the capabilities of GRUs to model sequential patterns and dependencies, we aim to develop a robust and accurate system able to differentiate between authentic domain names and those algorithmically generated for malicious intent. We will then describe the methodology used to train and evaluate the GRU model on a dataset of domain names, discussing the pre-processing steps, model architecture, and training procedures employed. Furthermore, we will present experimental results demonstrating the effectiveness of the GRU-based approach in accurately identifying algorithmically generated domain names. We will compare the performance of the GRU model with that of other machine learning approaches and baseline methods commonly used in the field of cyber security [6]. Ultimately, this research aims to contribute to the development of more robust and effective solutions for combating cyber threats posed by algorithmically generated domain names. By harnessing the power of advanced machine learning techniques like GRUs, we can enhance the security and resilience of internet infrastructure and protect users from malicious activities online [10].

Among the various architectures of RNNs, the GRU stands out as a dominant tool for sequential data processing. Unlike traditional RNNs, GRUs is designed to mitigate the vanishing gradient problem [12] and capture long-range dependencies in sequential data more effectively. These properties make GRUs well-suited for the task of analyzing domain names, which inherently possess sequential characteristics. The objective of this research paper is to conduct a comprehensive performance evaluation of deep learning models, including RNN, LSTM, and GRU architectures, for DGA generated domain name classification and detection [11]. By systematically comparing the effectiveness of these models, we aim to provide valuable insights into their strengths, limitations, and suitability for practical cyber security applications. In this research paper, we aim to address this gap by conducting a systematic evaluation of RNN, LSTM, and GRU models for DGA generated domain name classification and detection. Our contributions can be summarized as follows:

(1) We analyse the existing deep learning based DGA domain detection methods in detail
(2) We extend DGA domain detection from binary classification to multiclass classification to determine the type of DGA domain, not just whether it is a DGA domain
(3) We design and implement DGA domain detection methods using existing and proposed models and highlight the supremacy of the proposed methods through extensive performance comparison using the latest and most realistic datasets.
This paper evaluates effectiveness of large-scale deep learning approaches such as recurrent neural network (RNN), long short-term memory (LSTM), Gated Recurrent Unit (GRU) architectures to DGA classification. These deep networks are composed of complex units and inner mechanics of these units is remained as a black box. Thus an adversary may not be able to reverse engineer the classifier without knowing the same training samples. The increasing sophistication of cyber threats, particularly those leveraging Domain Generation Algorithms (DGAs), presents a significant challenge to cyber security professionals worldwide. DGAs are commonly used by malicious actors to dynamically produce a substantial quantity of domain names [11], thereby evading traditional detection mechanisms and facilitating various cyber attacks such as malware propagation, botnet command and control, and phishing. Detecting and mitigating the threats posed by DGA-generated domain names requires robust and adaptive detection mechanisms capable of effectively

distinguishing between malicious and legitimate domains in real-time.

The remainder of this paper is structured as follows: In Section 2. The related work on DGA domain detection is presented. In Section 3, the theoretical background of deep learning models is explained. In Section 4, the two proposed efficient detection methods in detail with existing models are presented. In Section 5, the overall procedure of DGA domain detection is described, and in Section 6, the performance evaluation of deep learning techniques are experimentally evaluated [21]. Finally, in Section 7, the conclusions of the study are presented

II. RELATED WORK

The realm of Domain Generation Algorithm (DGA) detection has seen extensive research efforts, with various approaches aimed at effectively identifying and mitigating the threat posed by DGAs. Some researchers have focused on analyzing the temporal and spatial characteristics of DNS traffic, utilizing features such as network and zone information to calculate reputation scores for individual domain names. The literature on deep learning models for DGA generated domain name classification and detection highlights the growing interest in leveraging RNN, LSTM, and GRU architectures for this task. Previous studies have demonstrated the effectiveness of these models in accurately identifying DGA-generated domain names and distinguishing them from legitimate domains. Previous research has explored various deep learning approaches for DGA domain detection, focusing on accuracy and robustness. However, there is a lack of comprehensive performance evaluations comparing the relative strengths and limitations of each model architecture. Our research aims to fill this gap by conducting a systematic evaluation of RNN, LSTM, and GRU models, providing insights into their performance characteristics and practical applicability in cyber security.

In their study, J. Woodbridge et al. [5] introduced a novel approach to predict Domain Generated Algorithms (DGAs) using Long Short-Term Memory (LSTM) networks. Their research focused on developing a DGA classifier that utilizes LSTM networks to accurately predict DGAs without the need for contextual knowledge or manually created features. One of the key strengths of their proposed method is its ability to perform multi-class classification effectively. This means that the classifier can not only identify whether a domain is generated by a DGA, but it can also assign it to a specific DGA family, yielding valuable insights into the nature of the threat. By harnessing the capabilities of LSTM networks, which are well-suited for sequence prediction tasks due to their ability to capture long-term dependencies, Woodbridge et al. achieved high accuracy in DGA detection. Their approach represents a significant advancement in the field of cyber security, offering a more automated and efficient method for identifying and categorizing DGAs, thus improving the capacity to defend against sophisticated cyber threats.

David Dagon et al. [10] this work investigates the effectiveness of different deep learning architectures, including LSTM and GRU networks, for detecting DGA-generated domain names. The authors experiment with various feature representations and model configurations to optimize detection performance. The results suggest that LSTM and GRU networks achieve superior performance compared to traditional RNNs, with LSTM slightly outperforming GRU in some scenarios.

The native class imbalance of DGA data was addressed by Tran et al. [11] while some researchers augmented their training data with additional known DGA datasets or incorporated more contextual information into their scoring mechanism. Another approach involved modifying the original LSTM architecture to a bidirectional LSTM layer, showcasing the potential improvements achievable through architectural changes. Addressing Class Imbalance the author Tran recognized the inherent class imbalance in DGA datasets, where certain DGA families may be much more prevalent than others. They likely employed techniques to mitigate this issue, such as oversampling minority classes or adjusting class weights during training, to ensure the model learns effectively from all classes.

Ryan R. Curtin [15] introduced an algorithm designed for the detection of Domain Generation Algorithm

(DGA) domains utilizing Recurrent Neural Networks (RNNs) alongside supplementary information. Through extensive experiments, Curtin demonstrated the efficacy of the model in accurately identifying domains generated by complex DGA families. Notably, the algorithm showcased superior performance compared to previous methods, particularly excelling in the detection of intricate DGA families such as rovnix, suppobox, and matsnu, among others. Moreover, Curtin's algorithm can serve as a standalone DGA domain detector, providing a valuable endpoint application for cyber security purposes. Alternatively, it can be seamlessly integrated as a component within a larger malware detection system, enhancing the overall efficacy of threat mitigation efforts. By leveraging RNNs and supplementary information, Curtin's algorithm represents a significant advancement in DGA detection technology, offering enhanced capabilities in identifying and mitigating complex cyber threats. Its versatility and effectiveness make it a valuable asset in the ongoing battle against malware and cyber-attacks.

The author Qiao [16] et al. was one among the pioneers in training deep learning models tailored for dictionary-based DGA domains. They employed a pre-trained embedding for the words within domain names and trained an LSTM on both single-DGA and multiple-DGA datasets. While their results set a benchmark for dictionary DGA detection, their model faced significant limitations due to its reliance on context-sensitive word embedding and the omission of some available data during training and testing. Nonetheless, Qiao work represents a notable advancement in the field of DGA detection, with a focused emphasis on dictionary-based DGA domains. Their approach, integrating pre-trained word embeddings with LSTM networks, showcased substantial progress in identifying this specific type of DGA.

The author Vaibhav Shah et al. [18] have compared the performance of convolution neural network CNN and RNN models, including LSTM and GRU variants, for detecting DGA-generated domain names. The models are trained and evaluated on a diverse dataset, with results demonstrating that both LSTM and GRU networks outperform basic RNNs in terms of detection accuracy and robustness, with LSTM showing slightly better performance than GRU.

Mohammad Reza, et al. [19] introduced a deep learning-driven method for identifying domain generation algorithm (DGA) malware. This study compares the performance of LSTM and GRU networks for detecting domain generation algorithm (DGA) malware. The models are trained on a dataset of DGA and benign domain names, with results indicating that both LSTM and GRU networks achieve high detection accuracy. However, LSTM exhibits slightly better performance in terms of precision and recall compared to GRU.

Athira C K et al. [20] this study evaluates the performance of different deep learning architectures, including CNN-LSTM and CNN-GRU models, for DGA domain detection. The models are trained on a dataset of DGA and legitimate domain names, with results indicating that both LSTM and GRU variants achieve high detection accuracy. However, the CNN-GRU model demonstrates marginally superior performance in terms of both accuracy and computational efficiency. The authors incorporate contextual information to enhance detection accuracy and compare the models' performance using metrics such as accuracy and F1-score

Another study focuses on enhancement of RCNN Liu et al. [22] present an enhancement to the Recurrent CNN proposed by Lai et al. [19]. Their modification, known as the Recurrent Convolution Neural Network with Spatial Pyramid Pooling (RCNN-SPP), involves refining the pooling algorithm within the convolution layers to enhance the representation of domain name features. By employing multiple filters of varying sizes, the RCNN-SPP captures a diverse range of feature representations, which are then integrated to form the final representation. This adaptation achieves an impressive accuracy of 92% for both binary and multi-class classification tasks, demonstrating its effectiveness in DGA detection.

Yang et al. [23] introduce a Heterogeneous DNN, comprising two distinct models. The first component, an Improved Parallel CNN (IPCNN), employs multiple CNNs with different kernel sizes to extract local features across various scales. The second

component, a Self-Attention based Bi-LSTM (SA-Bi-LSTM), focuses on capturing global features. Finally, the IPCNN and SA-Bi-LSTM layers are combined to provide a comprehensive classification outcome. Through experimentation, HDNN outperforms previous approaches on the evaluated dataset, showcasing its superiority in DGA detection tasks.

Shibahara et al. [26] introduced a modified algorithm that employs Recurrent Neural Networks (RNNs) to analyze variations in network communication, resulting in a significant reduction in malware analysis time. Unlike techniques specifically tailored to Domain Generation Algorithms (DGAs), this approach is more general and aims to address various types of malware threats by examining their communication patterns. The adaptability of Shibahara et al.'s algorithm to a wide range of malware types underscores its potential as a comprehensive solution for malware detection and analysis. By leveraging RNNs to efficiently process and analyze network communication variations, the algorithm offers a promising approach to expedite threat detection and mitigate the impact of malicious activities on network security.

Several notable studies have investigated the efficacy of RNN, LSTM, and GRU architectures for DGA detection and classification. Gao et al. (2018) conducted a comparative study of deep learning models for DGA detection, highlighting the effectiveness of LSTM networks in capturing temporal dependencies within domain name sequences. Similarly, Zhang et al. (2019) proposed a hybrid deep learning framework combining Convolution Neural Networks (CNNs) and RNNs for DGA detection, achieving promising results across multiple DGA families. Moreover, Malware Hunter Team (2018) introduced a novel approach to DGA detection using GRU networks, demonstrating superior performance compared to traditional machine learning techniques. These studies underscore the importance of leveraging deep learning models, particularly RNNs, LSTMs, and GRUs, in effectively addressing the challenges posed by DGAs in cyberspace.

III. DGA (DOMAIN GENERATION ALGORITHM)

DGAs are commonly utilized in Command and Control (C&C) servers for botnets and ransom ware. Blocking these servers can disrupt the operations of threat actors, preventing them from communicating with infected machines. The constant rotation of domains associated with C&C servers is known as Domain Fluxing or Fast Fluxing. DGAs are integral to modern malware, enabling attackers to bypass security measures and evade detection. Domain Generation Algorithms (DGAs) play a crucial role in the evolution of malware, serving as a method for generating malicious domain names using algorithms. In the past, malware would often use hardcoded domain names or IP addresses to establish connections with botnets. However, DGAs have emerged as a more sophisticated approach, enabling malware to dynamically generate a large number of random domains.

The primary objective of DGAs is to create a pool of potential domains from which the botnet operator registers one or a few domains to activate the malware. Should a registered domain be detected and blocked, the algorithm can generate a fresh domain to maintain control of the botnet. The purpose of developing DGA classifiers is not solely to take down or block botnets but to detect and identify their presence within systems or services. DGAs vary in complexity, ranging from simple algorithms that consistently produce domain names to more advanced models that mimic the distribution patterns observed in legitimate domains. In summary, DGAs represent a significant challenge in cyber security, facilitating the operations of malicious actors and enabling the persistence of malware threats. Developing effective DGA classifiers is essential for detecting and mitigating these evolving threats in order to protect systems and services from exploitation.

IV. DEEP LEARNING MODELS

Deep learning involves the process of learning hierarchical representations of data by utilizing architectures with multiple hidden layers. With the advancement of high-performance computing facilities, deep learning techniques using deep neural

networks have gained increasing popularity. In a deep learning algorithm, data is passed through multiple layers, with each layer progressively extracting features and transmitting information to the subsequent layer. The initial layers extract low-level characteristics, which are then combined by later layers to form a comprehensive representation.

In traditional machine learning techniques, the classification task typically involves a sequential process that includes pre-processing, feature extraction, meticulous feature selection, learning, and classification. The effectiveness of machine learning methods heavily relies on accurate feature selection, as biased feature selection can lead to incorrect class classification. In contrast, deep learning models enable simultaneous learning and classification, eliminating the need for separate steps. This capability makes deep learning particularly advantageous for automating feature learning across diverse tasks. In the era of deep learning, a wide array of methods and architectures has been developed. These models can be broadly categorized into two main groups: discriminative (supervised) and generative (unsupervised) approaches. Among the discriminative models, two prominent groups are convolution neural networks (CNNs) and recurrent neural networks (RNNs).

1) Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNNs) are a class of deep learning models that possess internal memory, enabling them to capture sequential dependencies. Unlike traditional neural networks that treat inputs as independent entities, RNNs consider the temporal order of inputs, making them suitable for tasks involving sequential information. By employing a loop, RNNs apply the same operation to each element in a series, with the current computation depending on both the current input and the previous computations. The structure of an RNN, as depicted in Figure 2, allows it to capture important information from past data, enabling accurate predictions of future data points. This capability makes RNNs particularly effective for tasks involving sequential data such as speech recognition, time series analysis, financial forecasting, natural language processing, and more.

In this context, RNNs offer several advantages over traditional detection methods. They can adapt to changes in DGA techniques and evolve alongside emerging threats, providing a more robust defense against malicious activities. Additionally, RNNs have the potential to enhance the efficiency and accuracy of DGA detection systems, ultimately bolstering cyber security defenses and safeguarding networks from malicious intrusions. RNN has emerged as a valuable tool for detecting Domain Generation Algorithms (DGAs) within the cyber security domain. RNNs are particularly well-suited for this task due to their ability to capture sequential patterns in data, making them effective at analysing the complex sequences of characters present in DGA-generated domain names.

The architecture of an RNN includes loops within the network that allow information to persist over time, enabling the model to consider the entire sequence of characters when making predictions. This characteristic is crucial for DGA detection, as it allows the model to capture the nuanced patterns and dependencies present in domain names generated by DGAs. In practice, RNNs for DGA detection are trained on large datasets containing labeled examples of both legitimate and malicious domain names. During training, the network learns to automatically extract relevant features from the input data and classify domain names as either legitimate or DGA-generated.

One common challenge in using RNNs for DGA detection is the tendency for gradients to vanish or explode during training, which can hinder the model's ability to learn long-range dependencies. However, techniques such as gradient clipping and gated architectures like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks have been developed to address these issues and improve the performance of RNNs in sequence modeling tasks. Overall, RNNs offer a powerful and flexible framework for DGA detection, enabling cyber security professionals to effectively identify and mitigate the threat posed by DGAs in network traffic. As with any machine learning technique, the effectiveness of RNNs for DGA detection depends on factors such as the quality and diversity of the training data, as well as the specific characteristics of the DGA families being targeted.

The ability of RNNs to utilize contextual information is particularly valuable in tasks such as natural

language processing, video classification, and speech recognition. For example, in language modeling, understanding the preceding words in a sentence is crucial for predicting the next word. RNNs excel at capturing such dependencies due to their recurrent nature.

However, a limitation of simple RNNs is their short-term memory, which restricts their ability to retain information over long sequences. To overcome this, more advanced RNN variants have been developed, including Long Short-Term Memory (LSTM), bidirectional LSTM, Gated Recurrent Unit (GRU), bidirectional GRU, Bayesian RNN and others. Figure 1 depicts a simple recurrent neural network, where the internal memory (ht) is computed using Equation (1)

$h_t = (Wxt + Uht + b)$ (1)

In this equation, (1) represents the activation function (typically the hyperbolic tangent), U and W are adjustable weight matrices for the hidden state (h), b is the bias term, and x denotes the input vector.

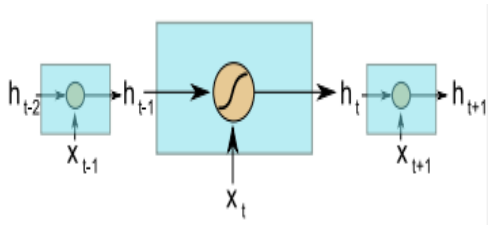

Fig. 1. Simple RNN internal operation.

2) Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is an advanced variant of Recurrent Neural Networks (RNN) that addresses the issue of capturing long-term dependencies. LSTM was initially introduced in 1997 and further improved in 2013, gaining significant popularity in the deep learning community. Compared to standard RNNs, LSTM models have proven to be more effective at retaining and utilizing information over longer sequences. Unlike traditional RNNs, LSTMs excel at retaining information over extended periods, making them ideal for processing and analyzing time series data where indefinite delays and complex dependencies are common. LSTMs were specifically engineered to mitigate issues such as the vanishing and exploding gradient problems encountered during training of conventional RNNs. One notable advantage of LSTMs is their robustness in identifying patterns within large sequences, such as those found in speech and text data. In the context of DGA detection, LSTMs can be leveraged to learn and recognize patterns in sequences of characters, such as domain names, enabling the classification of domains as either DGA-produced or legitimate.

Detecting Domain Generation Algorithms (DGAs) poses a formidable challenge in cyber security, given their role in facilitating the operation of botnets and malware. Traditional methods often struggle to effectively identify and mitigate the threat posed by DGAs due to their dynamic and evolving nature. In recent years, Recurrent Neural Networks (RNNs) have emerged as a promising approach for DGA detection, leveraging their ability to capture sequential patterns in data. By utilizing LSTMs for DGA detection, researchers can harness the network's ability to capture intricate patterns and dependencies within sequential data, thereby enhancing the accuracy and effectiveness of DGA detection systems

The architecture of an LSTM network includes memory cells that can retain information over extended periods, allowing them to effectively capture the complex relationships and dependencies within sequences of characters, such as domain names. This capability enables LSTMs to discern subtle patterns indicative of DGA-generated domains, even in the presence of noise or variations. In practice, LSTM networks are trained on large datasets containing both legitimate and malicious domain names. During training, the network learns to automatically extract relevant features from the input data and classify domain names as either legitimate or DGA-generated. By leveraging the advanced capabilities of LSTM networks, cyber security professionals can enhance their ability to detect and mitigate the threat posed by DGAs, thereby bolstering the security of networks and systems against malware and botnet attacks.

In an LSTM network, the current input at a specific time step and the output from the previous time step is fed into the LSTM unit, which then generates an output that is passed to the next time step. The final hidden layer of the last time step, sometimes along with all hidden layers, is commonly employed for classification purposes. The overall architecture of an LSTM network is depicted in Figure 2.

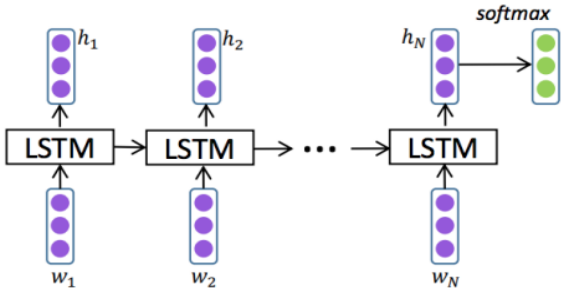

Fig. 2. The high- level architecture of LSTM model

LSTM consists of three gates: input gate, forget gate, and output gate. Each gate performs a specific function in controlling the flow of information. The input gate decides how to update the internal state based on the current input and the previous internal state. The forget gate determines how much of the previous internal state should be forgotten. Finally, the output gate regulates the influence of the internal state on the system. Figure 3 illustrates the update mechanism within the inner structure of an LSTM.

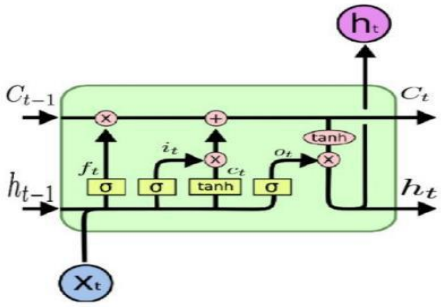

Fig. 3.The inner architecture of a standard LSTM module

3) Gated Recurrent Unit (GRU)

The Gated Recurrent Unit (GRU) is another variant of the RNN architecture that addresses the short-term memory issue and offers a simpler structure compared to LSTM. GRU combines the input gate and forget gate of LSTM into a single update gate, resulting in a more streamlined design. Unlike LSTM, GRU does not include a separate cell state.

GRU networks consist of gating mechanisms that control the flow of information within the network, allowing them to effectively capture and retain relevant information over time. This enables GRUs to discern subtle patterns and anomalies within sequences of characters, such as those found in DGA-generated domain names. In the context of DGA detection, GRU networks are trained on large datasets containing labeled examples of legitimate and malicious domain names. Through the process of training, the network learns to automatically extract meaningful features from the input data and classify domain names as either legitimate or DGA-generated. By leveraging the capabilities of GRU networks, cyber security professionals can enhance their ability to detect and mitigate the threat posed by DGAs, thereby strengthening the security posture of networks and systems against malware and botnet attacks. As with any machine learning technique, the effectiveness of GRUs for DGA detection depends on factors such as the quality and diversity of the training data, as well as the specific characteristics of the DGA families being targeted.

A GRU unit consists of three main components: an update gate, a reset gate, and the current memory content. These gates enable the GRU to selectively update and utilize information from previous time steps, allowing it to capture long-term dependencies in sequences. Figure 4 illustrates the structure of a GRU unit.

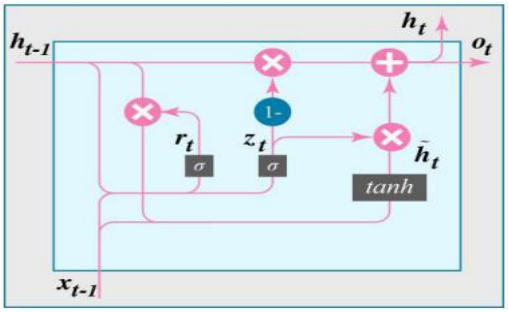

Fig. 4. the structure of a GRU unit.

The update gate (equation 2) determines how much of the past information should be retained and combined with the current input at a specific time step. It is computed based on the concatenation of the previous hidden state $ht-1$ and the current input xt, followed by a linear transformation and a sigmoid activation function.

$zt = ([ht-1,t]+bz)$ (2)

The reset gate (equation 3) decides how much of the past information should be forgotten. It is computed in a similar manner to the update gate using the

concatenation of the previous hidden state and the current input.

rt= ([ht−1,t]+br) (3)

The current memory content (equation 4) is calculated based on the reset gate and the concatenation of the transformed previous hidden state and the current input. The result is passed through a hyperbolic tangent activation function to produce the candidate activation.

$\tilde{h}t$=$tanh$ (Wh [$rtht$−1, xt]) (4)

ht = (1−zt) ht−1+$zt\tilde{h}t$ (5)

Finally, the final memory state ht is determined by a combination of the previous hidden state and the candidate activation (equation 5). The update gate determines the balance between the previous hidden state and the candidate activation. Additionally, an output gate ot can be introduced to control the information flow from the current memory content to the output. The output gate is computed using the current memory state ht and is typically followed by an activation function, such as the sigmoid function.

ot=($Woht$+bo) (5)

Where the weight matrix of the output layer is Wo and the bias vector of the output layer is bo.

GRU offers a simpler alternative to LSTM with fewer tensor operations, allowing for faster training. However, the choice between GRU and LSTM depends on the specific use case and problem at hand. Both architectures have their advantages and disadvantages, and their performance may vary depending on the nature of the task [26].

V. DATASET DESCRIPTION

The proposed domain name detection model was evaluated on Amrita DGA dataset for discovering malwares/botnets from the DNS traffic. Amrita DGA is a benchmark dataset publically available for research purpose. This database was used in DMD-2018 shared task and after the shared task this database has been used for benchmark purpose by various researchers for DGA detection. Following, in this work, the Amrita DGA database was used for DGA domains detection. The domain name in the dataset is labeled as benign or DGA family. The dataset is further divided into training and testing respectively.

The evaluation is conducted on a dataset comprising both DGA-generated domain names and legitimate domain names. The dataset is carefully curated to represent a diverse range of DGA families and legitimate domain naming conventions. It includes features such as domain name strings, character-level representations, and possibly additional contextual information relevant to DGA detection. Preprocess the dataset to normalize domain names, encode characters into numerical representations, and split the dataset into training, validation, and testing sets.

All deep learning models are trained using the training dataset. Further the dataset is comprised of training, validation and testing dataset. The training, validation and testing domain name samples are shown in the below

Tab. 1. Detailed information of the dataset

Label	Domain Type	Training	Testing	Validation
0	benign	25574	6414	8079
1	banjori	3779	1021	1165
2	corebot	3815	954	1242
3	dircrypt	3890	942	1201
4	dnschanger	3883	961	1187
5	fobber	3815	953	1203
6	murofet	3823	942	1237
7	necurs	3282	823	993
8	newgoz	3858	987	1185
9	padcrypt	3802	932	1145
10	proslikefan	3823	946	1176
11	pykspa	3834	940	1194
12	qadars	3848	972	1172
13	qakbot	3844	964	1209
14	ramdo	3907	963	1198
15	ranbyus	3868	980	1258
16	simda	3870	959	1195
17	suppobox	3850	948	1234
18	symmi	3802	952	1173
19	tempedreve	3818	932	1179

20	tinba	3845	973	1197

VI. PERFORMANCE METRICS

To evaluate the performance of these models, we employed assessment metrics such as accuracy, precision, recall, and F1-measure. Accuracy measures the overall correctness of the model's predictions, while precision evaluates the proportion of correctly predicted positive instances. Recall assesses the model's ability to correctly identify positive instances, and F1-measure provides a balanced measure of precision and recall. Evaluate the performance of each model on the testing set using standard metrics for classification tasks, such as accuracy, precision, recall, F1-score, and receiver operating characteristic (ROC) curve analysis. Additionally, analyze other relevant metrics such as detection rate, false positive rate, and area under the ROC curve (AUC) to assess the model's overall performance.

Accuracy in DGA detection and classification measures the overall correctness of the model's predictions regarding whether a domain is generated by a DGA or not.

$$\text{Accuracy} = \frac{\text{Total number of domains}}{(\text{Number of correctly classified domains})}$$

Precision in DGA detection and classification measures the accuracy of positive predictions made by the model. It calculates the ratio of correctly identified DGA-generated domains to the total domains predicted as DGA-generated by the model.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}$$

Recall in DGA detection and classification measures the ability of the model to identify all DGA-generated domains correctly. It calculates the ratio of correctly identified DGA-generated domains to all DGA-generated domains present in the dataset.

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}$$

F1-score in DGA detection and classification is the harmonic mean of precision and recall. It provides a balance between precision and recall, which is particularly useful when dealing with imbalanced datasets.

$$\text{F1} - \text{Score} = 2 \times \frac{(\text{Recall} * \text{Precision})}{(\text{Recall} + \text{Precision})}$$

These metrics are crucial for evaluating the performance of DGA detection and classification models. Given the malicious nature of DGA-generated domains, high precision and recall are typically desired to minimize false positives and false negatives, respectively.

$$\text{TPR} = \frac{\text{True Positive}}{\text{TruePositive} + \text{FalseNegative})}$$

$$\text{FPR} = \frac{\text{FalsePositive}}{(\text{FalsePositive} + \text{TrueNegative})}$$

$$\text{AUC} = \int_0^1 \frac{\text{TP}}{(\text{TP} + \text{FN})} \, \text{d} \frac{\text{FP}}{\text{TN} + \text{FP}}$$

VII. RESULTS AND DISCUSSION

In our experimental analysis, we utilized AmrithaDGA dataset. The purpose was to perform a performance analysis of various deep learning models. Specifically, we examined four different models: LR, RNN, LSTM, and GRU. By evaluating these models on the testing data, we aimed to assess their performance using multiple evaluation metrics such as accuracy, precision, recall, and F1-measure. The evaluation is conducted on a dataset comprising both DGA-generated domain names and legitimate domain names. The dataset is carefully curated to represent a diverse range of DGA families and legitimate domain naming conventions. It includes features such as domain name strings, character-level representations, and possibly additional contextual information relevant to DGA detection. Pre-process the dataset to normalize domain names, encode characters into numerical representations, and split the dataset into training, validation, and testing sets. Our analysis focused on comparing the performance of the different deep learning models across the dataset. Simple RNN (Recurrent Neural Network) is suitable for sequential data analysis, while LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) models excel in capturing long-term dependencies in sequential data. Bidirectional LSTM and Bidirectional GRU models offer the advantage of processing information in both forward and backward directions.

The results analysis chapter provides insights into the performance of RNN, LSTM, and GRU models for DGA generated domain name classification and detection. We compare the accuracy, precision, recall,

and computational efficiency of each model architecture, highlighting their relative strengths and limitations. Additionally, we assess the generalization ability and robustness of the models to emerging threats, providing valuable insights for cyber security practitioners and researchers.

Our results demonstrate that GRU architecture consistently outperforms traditional RNNs in DGA generated domain name classification and detection tasks. These models leverage their ability to capture long-range dependencies and overcome the vanishing gradient problem, leading to improved accuracy and robustness. We observed a trade-off between precision and recall across different model architectures. While GRU model exhibit higher recall rates, it may suffer from slightly lower precision compared to RNNs and LSTM. This trade-off highlights the importance of carefully balancing false positives and false negatives in DGA detection scenarios.

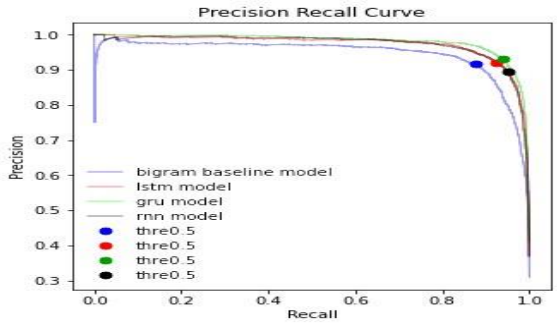

Fig. 5. Performance comparison of baseline model with deep learning models

The above Figure 5, the precision diagram, provides a visual representation of how the different deep learning models and baseline model bigram with logistic regression perform in terms of efficiency during the training process. PRC curve on the detection of DGA domain names. Whereas Fig 6 represents the comparison of various deep learning models. The x-axis indicates the false positive rate while the y-axis shows the true positive rate.

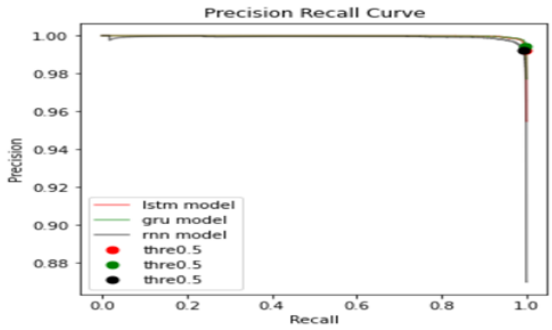

Fig. 6. Performance comparison of deep learning models

The comparisons of accuracy, precision, recall and F1 score for the deep learning methods are displayed in Fig 7, the gated recurrent unit model outperforms the other four models on the accuracy, precision, recall, and F1 score. The GRU model has better detection result on F1-score of 0.9880 than the deep learning models. The recall and precision of the GRU model are higher than those of the best performing LSTM and RNN model in the traditional method, respectively, indicating that the deep learning models can better detect DGA domain names and improve the overall detection effect.

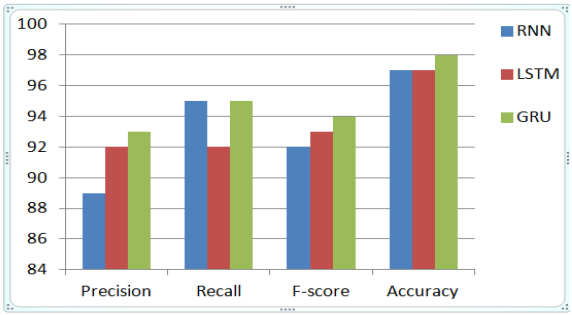

Fig. 7. Overall performance analysis of deep learning models for DGA domain detection

The performance comparison of baseline model logistic regression with the proposed deep learning models are measured in terms of performance evaluation metrics such as accuracy, precision, recall and F1-score.

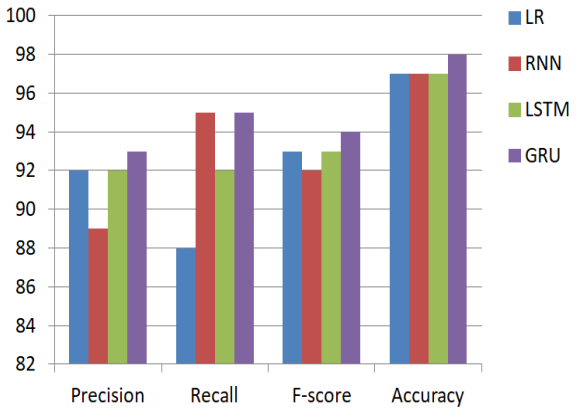

Fig. 8. Performance comparison of baseline model with deep learning models

The trained model's performance is assessed using testing samples at each epoch. LSTM exhibits strong performance up to 500 epochs, after which it begins to decline due to over fitting. Meanwhile, RNN's performance starts decreasing after 300 epochs, and GRU begins over fitting after just 100 epochs. RNN shows signs of over fitting as early as 50 epochs. This suggests that 500 epochs are adequate for capturing domain name dependencies at the character level. As a benchmark, we apply a logistic regression model to character-level bigrams of domain names. Experiment results in Table 2 display accuracy, precision, recall, and F1-score for classifying domain names as benign or DGA-generated. Table 3 reveals that GRU outperforms RNN, LSTM, and other methods in both binary and multi-class classification scenarios.

Tab. 2. Summary test results for binary classification

Algorithm	Precision	Recall	F-score	Accuracy
Bigram-LR	92	88	93	97
RNN	89	95	92	97
LSTM	92	92	93	97
GRU	93	95	94	98

Tab. 3. Summary of test results for multi-class classification

	Bigram with LR			RNN			LSTM			GRU		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
benign	0.98	0.99	0.99	0.99	0.94	0.97	0.99	0.99	0.99	0.99	0.99	0.99
banjori	1.00	1.00	1.00	0.98	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
corebot	1.00	1.00	1.00	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
dircrypt	0.52	0.57	0.54	0.51	0.46	0.48	0.67	0.68	0.68	0.71	0.70	0.71
dnschanger	0.57	0.85	0.68	0.49	1.00	0.66	0.90	0.95	0.93	0.90	0.97	0.94
fobber	0.71	0.94	0.81	0.75	0.99	0.86	0.87	0.94	0.90	0.88	0.95	0.91
murofet	0.96	0.95	0.96	0.80	0.89	0.84	0.90	0.94	0.92	0.93	0.94	0.94
necurs	0.88	0.71	0.79	0.90	0.58	0.70	0.94	0.85	0.90	0.96	0.85	0.90

newgoz	1.00	0.99	1.00	0.98	0.98	0.98	1.00	1.00	1.00	1.00	1.00	tempedreve	0.56	0.39	0.46	0.76	0.70	0.73	0.71	0.76	0.73	0.76	0.70	0.73
padcrypt	1.00	1.00	1.00	0.74	0.77	0.76	0.99	0.99	1.00	1.00	0.99	tinba	0.80	0.84	0.82	0.87	0.65	0.75	0.94	0.93	0.93	0.92	0.96	0.94
proslikefan	0.78	0.58	0.66	0.67	0.60	0.63	0.81	0.64	0.71	0.72	0.71	accuracy			0.87			0.83			0.92			0.92
pykspa	0.55	0.58	0.57	0.49	0.62	0.55	0.68	0.80	0.73	0.68	0.78	macro avg	0.84	0.84	0.83	0.80	0.79	0.78	0.90	0.90	0.90	0.91	0.90	0.90
qadars	1.00	1.00	1.00	0.97	0.97	0.97	1.00	0.99	0.99	1.00	1.00	weighted avg	0.87	0.87	0.87	0.84	0.83	0.82	0.92	0.92	0.92	0.92	0.92	0.92
qakbot	0.65	0.47	0.54	0.50	0.28	0.36	0.65	0.54	0.59	0.70	0.58	0.63												
ramdo	0.98	1.00	0.99	0.90	0.98	0.94	1.00	1.00	1.00	1.00	1.00	1.00												
ranbyus	0.79	0.72	0.75	0.81	0.76	0.78	0.89	0.86	0.87	0.89	0.88	0.88												
simda	0.99	0.99	0.99	0.92	0.97	0.95	0.99	1.00	1.00	1.00	0.99	1.00												
suppobox	0.97	1.00	0.99	0.89	0.84	0.86	0.98	0.99	0.99	0.98	0.99	0.99												
symmi	1.00	1.00	1.00	0.98	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00												

CONCLUSION

In this study, we analysed deep learning-based DGA domain detection methods. The results demonstrate that the CNN model performs exceptionally well in image classification tasks, while the RNN models, such as LSTM and GRU, show strong performance in time series analysis. Additionally, the GRU model stands out for its faster training time compared to LSTM, attributed to its simplified architecture with two gates instead of three. This study assesses the efficacy of various deep learning methodologies in detecting and categorizing domain names associated with specific malware families, generated by domain generation algorithms (DGAs). By training on character-level data and automatically extracting relevant features, the models accurately differentiate between malicious and benign domain names. Future research should explore the effectiveness of deep learning techniques across a broader spectrum of malware families. Furthermore, while this paper does not delve into the inner workings of deep learning networks, understanding these mechanics is crucial for real-time deployment. Thus, future investigations should focus on analysing the internal mechanisms of deep learning algorithms, emphasizing their detection

rates for malicious domain names and providing suitable mathematical representations and visualizations.

REFERENCES

[1] S. Yadav, A. K. K. Reddy, A. Reddy, and S. Ranjan, “Detecting algorithmically generated malicious domain names,” in Proc. 10th ACM SIGCOMM conference on Internet measurement, pp. 48–61, ACM, 2010.

[2] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, “Detecting algorithmically generated domain-flux attacks with DNS traffic analysis,” Networking, IEEE/ACM Transactions on, vol. 20, no. 5, pp. 1663–1677, 2012.

[3] M. Antonakakis et al., “From Throw-Away Traffic to Bots: Detecting the Rise of DGABased Malware,” in Proceedingsof the 21st USENIX conference on security symposium (Security’12), 2012.

[4] A. Karim, R. Bin Salleh, M. Shiraz, S. A. A. Shah, I. Awan, and N. B. Anuar, “Botnet detection techniques: review, future trends, and issues,” Journal of Zhejiang University: Science C, vol. 15, no. 11, pp. 943–983, 2014, doi: 10.1631/jzus.C13002

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

[6] P. Lison and V. Mavroeidis, "Automatic detection of malware generated domains with recurrent neural models", arXiv preprint arXiv:1709.07102, 2017.

[7] Z. Feng, C. Shuo, and W. Xiaochuan, “Classification for DGAbased malicious domain names with deep learning architectures,” Int’l Journal of Intelligent Information Systems, vol. 6, no. 6, pp. 67–71, 2017.

[8] Yonatan Bozorgy, Daniel Barthel, Taejoong and Kim "Detecting DGA Domains with Recurrent Neural Networks and Side Information" International Conference on Artificial Intelligence and Security (AISec), 2017

[9] J. Saxe and K. Berlin, “expose: a character-level convolutional neural network with embeddings for detecting malicious urls,” File Paths, and Registry Keys, vol. 34, 2017.

[10] David Dagon, Yoonjoon Lee, Juan Castro Fernandez "Deep Learning Based DGA Detection" Published in Proceedings of the First Workshop on Machine Learning and Computer Security (MLCS), 2018.

[11] 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.

[12] Vinayakumar, R., Soman, K.P. and Poornachandran, P., “Detecting malicious domain names using deep learning approaches at scale”. Journal of Intelligent & Fuzzy Systems, 34(3), pp.1355-1367, 2018.

[13] Vinayakumar, R., Soman, K., Poornachandran, P., Sachin Kumar, S.: Evaluating deep learning approaches to characterize and classify the DGAs at scale. J. Intell. Fuzzy Syst. 34(3), 1265–1276 (2018)

[14] Bharathi, B., and J. Bhuvana. “Domain Name Detection and Classification Using Deep Neural Networks”. In International Symposium on Security in Computing and Communication, pp. 678-686. Springer, Singapore, 2018.

[15] Curtin, R. R., Gardner, A. B., Grzonkowski, S., Kleymenov, A., & Mosquera, A. “Detecting DGA domains with recurrent neural networks and side information”. In Proceedings of the 14th International Conference on Availability, Reliability and Security (pp. 1-10) August 2019.

[16] 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,” Applied Sciences, vol. 9, no. 20, 2019.

[17] A. Shewalkar, D. Nyavanandi, and S. A. Ludwig, "Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU," Journal of Artificial Intelligence and Soft Computing Research, vol. 9, no. 4, pp. 235-245, 2019.

[18] Vaibhav Shah, Dr. S.S. Agrawal "DGA-Domain Detection Using CNN and RNN Model" Published in: 2019 International Conference on Vision towards Emerging Trends in Communication and Networking (ViTECoN), 2019

[19] Mohammad Reza Kangavari, Hidayet Aksu, Sevil Sen "Detection of Domain Generation Algorithm Malware Using Deep Learning"Published in: 2020 International Conference on Engineering and Emerging Technologies (ICEET), 2020

[20] Athira CK, Kavya Manohar, Soman KP "Effective DGA Domain Detection using Deep Learning and Contextual Information" Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), 2020

[21] J. Xiao and Z. Zhou, "Research progress of RNN language model," in 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), 2020: IEEE, pp. 1285-1288.

[22] Z. Liu, Y. Zhang, Y. Chen, X. Fan and C. Dong, "Detection of algorithmically generated domain names using the recurrent convolutional neural network with spatial pyramid pooling", Entropy, vol. 22, no. 9, pp. 1058, Sep. 2020.

[23] L. Yang, G. Liu, Y. Dai, J. Wang and J. Zhai, "Detecting stealthy domain generation algorithms using heterogeneous deep neural network framework", IEEE Access, vol. 8, pp. 82876-82889, 2020.

[24] W. Fang, Y. Chen, and Q. Xue, "Survey on research of RNN-based spatio-temporal sequence prediction algorithms," Journal on Big Data, vol. 3, no. 3, p. 97, 2021.

[25] Performance Analysis of DGA-Driven Botnets using Artificial Neural networks Manikanda Devaraja N, R. D, Seshadri Raghavendra Murali., Vandana Sharma 13 Oct 2022-pp 1-6

[26] Toshiki Shibahara, Takeshi Yagi “Efficient Dynamic Malware Analysis Based on Network Behavior Using Deep Learning” GLOBECOM 2022 - 2022 IEEE Global Communications Conference December 2022