PANGA: Attention-based Principal Neighborhood Aggregation for Forecasting Future Cyber Attacks

Alok Kumar Trivedi
Department of Computer Science
Indian Institute of Technology, Kanpur
alokt21@iitk.ac.in

Priyanka Bagade
Department of Computer Science
Indian Institute of Technology, Kanpur
pbagade@iitk.ac.in

***Abstract*—There has been a significant spike in cyber attacks with serious economic, security, and privacy concerns recently. The effectiveness of older NIDS (Network Intrusion Detection Systems) has been diminishing as cyber attacks become more sophisticated and complex. Majority of research focuses on detecting and classifying attacks, with limited work attempting to forecast the number of cyber attack. Recent works propose using statistical, machine learning, or deep learning models to forecast cyber attacks, but they do not take advantage of the correlation that exists among different devices present in a network. We proposed a model, PANGA that employs a combination of Principal Neighborhood Aggregation(PNA), Gated Recurrent Unit(GRU), and attention layers to effectively capture and leverage spatio-temporal dependencies. It achieves a mean square error of 0.027 and a coefficient of determination of 0.87 on well known CIDDS-001 dataset. Additionally, we performed perturbation analysis by adding gaussian noises to the test data to validate the robustness of the model. The performance of the proposed model remains largely unaffected as long as the standard deviation of the noise was kept below 0.25.**

***Index Terms*—Cyber attack forecasting, spatio-temporal, GNN, GRU**

I. Introduction

In today's world, the threat of cyber attacks presents a severe problem to the security and stability of various institutions ranging from financial and healthcare infrastructure[1]. With the growth of newly developed and sophisticated cyber-attacks, the preventive measures to detect them have become obsolete in various sector like business, government and private organizations[2]. When compared to the same period in 2022, the weekly frequency of such attacks increased by 7% in Q1 2023 [3]. The estimated global cost to prevent cyber attacks is forecasted to increase by 5.7 trillion US dollars between 2023 and 2028 [4]. Forecasting cyber attacks can help an individual or organization to take proactive measures and allocate necessary resources in advance to mitigate or minimize the impact of the attack.

There has been much work in the past dedicated to detecting and classifying cyber attacks, but a limited focus was given in forecasting a future attack [5] [6]. There are different types of forecasting techniques such as time series forecasting, and causal forecasting, quantitative forecasting etc. In this work we employ time series forecasting to predict future cyber attack as it enables the study of inherent temporal patterns present in the cyber attack dataset, analyze recurring attack pattern and cyclic behaviour. There are two types of time series forecasting: long-term and short-term. Long-term forecasting involves forecasting for an extended period into the future. The time range could be anywhere between several months to years. Short term forecasting involves making predictions for the near future time horizon. The time frame for short-term forecasts could range between a few minutes to hours. In this work, we use short term forecasting to predict attacks 15, 20, and 30 minutes ahead of time.

Traditional methods of cyber attack forecasting have primarily relied on statistical models like ARIMA [7], Vector Autoregression (VAR) [8], and Bayesian Structural Time Series (BSTS)[9], which often struggle to capture the evolving nature and complexity of modern cyber threats. Compared to them, deep learning models like GRU and LSTM are more equipped to handle nonlinear relationships and can capture long-term dependencies. However these models fail to realize the spatial properties of the dataset. They treat each flow of data independently and do not take advantage of the relationship that might exist between different data flows. Additionally, GRU/LSTM models are sensitive to the input sequence's order, which might affect the model's ability to capture long-term dependencies.

In this paper, we propose a novel forecasting model PANGA. It leverages the strengths of three key components: Principal Neighborhood Aggregation(PNA) [10], Gated Recurrent Units (GRUs), and an attention mechanism. PNA is used to represent the network structure as a graph, enabling the aggregation of information from neighboring nodes. By incorporating GRU, our model effectively captures temporal dependencies within attack sequences, discerning patterns and anomalies. Additionally, an attention mechanism enhances forecasting by selectively focusing on relevant information while filtering out irrelevant features, enhances the model's capability to identify attack indicators and generate accurate predictions. To our knowledge, this paper is the first work where we have attempted to forecast cyber attacks using GNN and attention mechanisms. We tested our model performance using the CIDDS-001 dataset[11]. Following are contributions of this work:

- We train a novel architecture consisting of PNA, GRU, and attention mechanism to forecast cyber attacks. The proposed PANGA model takes the spatiotemporal aspect of the dataset into account to forecast cyber attacks.
- We evaluate the performance of the proposed model on real-world cyber attack CIDDS-001 dataset [11].

The rest of the paper is organized as follows: We discuss existing works in cyber attack forecasting tasks and the different methodologies in section II. Section III discusses the problem statement, In section IV we present the architecture and components of our proposed PANGA model in detail. Section V describes the dataset, graph building process and data preprocessing. section VI explains the evaluation metrics, hyperparameters used in the model, analyses the result obtained and performance of the perturbation analysis. We conclude our work in section VII.

II. Related Work

This section discusses the state of the art models to predict cyber attacks. Werner et al. applied Arima to forecast the number of cyber attacks [12] on [13] dataset with 613 attacks recorded in 2016. They initialized their model with the first month's attack data and predicted the next day's attack number. They include the recorded number of attacks and update the model. They compared their model forecasting performance with the average value of past data. The model performance was considerably better than the mean value of the past data. Few of the works [14], [15] involved statistical and machine learning models to forecast cyber attacks. Autoregressive Integrated Moving Average(Arima) [7], Autoregressive Integrated Moving Average with Exogenous variables(Arimax), and the Hidden Markov Model [16] was implemented by [14] to forecast malicious attacks on the target system. The Arima model uses autoregressive (AR), differencing (I), and moving average (MA) components to handle temporal dependence and trends in the dataset. Arimax model, apart from taking past historical value into account, also considers external variables which may impact the target value for time series forecasting. They tested their models on the customized 1459 Cerber attack datasets. Arimax performed the best with 1.66, 2.1, 0.9 MAE, RMSE, and MASE, respectively. Likewise, Bakdash et al. [15] experimented with Bayesian State Space Model(BSSM)[17] and statistical models like ARMA and ARIMA. The dataset had 9302 instances of the cyber attack obtained from a large DoD Computer Security Service Provider(CSSP). They found that the machine learning model BSSM was able to perform better than statistical models like ARMA and ARIMA as the data has multiple cases of overdispersion and bursts, and the BSSM model was better able to accommodate those cases to predict cyber attacks one week ahead. BSSM obtained a Mean Absolute Percentage Error (MAPE) of 68.17 % for the entire data, 42.84 %, and 57.01 % for level three bursts and level two bursts data, respectively.

Some of the research works leveraged deep learning models to forecast cyber attacks. LSTM and GRU models are widely trained on weather and financial data to forecast future events. They can handle sequential data and can capture long-term dependencies. A BRNN-LSTM model on a honeypot dataset consisting of 166 IP addresses collected between 2010 and 2011 was trained by [18]. They created five different datasets by considering five distinct periods. The BRNN-LSTM model achieves an error rate of less than 5% for the I, II, III, and V datasets but could not perform well for the IV dataset. They also experimented with pure statistical methods like ARIMA and ARIMA + GARCH and a hybrid model, which first uses ARIMA to extract linear features from the dataset. Then they used a nonlinear approach like SVM or random forest to determine the nonlinear relationship. They found that the hybrid approach performed better for all five datasets than the statistical forecasting methods. K Albulayhi, QA Al-Haija implemented the ANN model with two hidden layers to forecast the malware and ransomware attacks[19]. They trained their model with time series worldwide ransomware and malware attack data collected between 2016 to 2021 and forecasted the ransomware and malware attack for the next five years (2022 to 2026). They achieved an accuracy of 0.99. A Bayesian Long Short-Term Memory(B-LSTM) model was trained to forecast large-scale cyber threats, years in advance[20]. Authors composed a dataset using Hackmageddon [13] website, Elsevier, Twitter, and Python API. The dataset covers major cyber incidents from 36 countries during the previous 11 years.

Above mentioned projects have used statistical methods, machine learning, and deep learning models for the cyber attack forecasting task. These models do not take into account the spatial correlation among the network devices, also none of them has used attention mechanisms to improve the performance of their model. There have been some works dedicated to traffic speed forecasting tasks where both GNN and attention mechanism were used [21], [22]. The authors[21] uses an encoder-decoder architecture with a transform attention layer in between to predict traffic speed. The encoder consists of a stacked ST-attention layer. ST-attention layer applies spatial attention and then temporal attention to thoroughly capture the spatiotemporal feature of the data. A novel transform attention layer was proposed between the encoder and decoder and the building block of the decoder is the LSTM layer. Chen et al. proposed a novel deep learning model called ASTGCN for traffic prediction [23]. The model incorporated graph convolutions and temporal attention mechanisms to capture dynamic spatial-temporal features of traffic data. They used a spatial-temporal convolution module to extract spatiotemporal features, a spatial-temporal attention module to assign weights to different features of the input, and a fully connected layer to map extracted features to output. In this paper, our proposed architecture utilizes PNA [10] to capture the spatial correlation between the network devices and GRU to handle the temporal dependencies present between the different temporal representations of the graph. The attention layer enables the model to filter out irrelevant parts of the input data while focusing on the relevant parts of input data by assigning weights to each part of the input data.

III. Problem Statement

In this work, we attempt to forecast cyber attacks for a certain period in the future by training the model on past data. We converted the tabular data into a directed weighted graph $G = (V, E)$ by treating each IP address as nodes belonging to set V and data flow between the nodes as edges and belonging to set E. $A \in R^{N*N}$ is the weighted unsymmetric adjacency matrix, where A_{ij} represent the strength of edge between the node i and j. $X \in R^{T*N*M}$ is the feature matrix, where it stores the feature vector of shape M for N nodes of the graph for T periods.

For a given $X \in R^{T_p*N*M}$, which stores cyber attack information for all the nodes of the graph over the past T_p time steps, our aim is to predict $Y \in R^{T_q*N}$, which stores the number of cyber attacks occurred for each node of the graph for the next T_q time step.

IV. Methodology

This section explains the proposed model PANGA architecture in detail. The model utilizes PNA, GRU and attention layers. We start with explaining basic of each of these layer and finally illustrates how all can be combined together to get better forecasting result with the proposed PANGA model.

A. PNA

Principal neighborhood aggregation(PNA)[10] works on the principle that in order to discriminate between a multi-set of size n, we need at least n aggregators. It uses four permutation invariant aggregators: mean, max, median, and standard deviation, along with degree scalars, to determine the node embeddings of the graph. PNA algorithm also uses higher moments if the degree of a node is high and four aggregators cannot describe the neighborhood correctly. Other GNN architectures, such as GCN[24] and GAT[25], use only one aggregator, which may not be enough to extract the valuable node information and may limit the learning capabilities of the GNN model.

The aggregators used in PNA architecture are mean, maximum, minimum, and standard deviation.

$$\mu_i(X^l) = \frac{1}{d_i} * \sum_{j \in N(i)} X_j^l \tag{1}$$

$$\mathrm{max_i}(X^l) = max_{j \in N(i)} X_j^l \tag{2}$$

$$\mathrm{min_i}(X^l) = min_{j \in N(i)} X_j^l \tag{3}$$

$$\sigma_i(X^l) = \sqrt{\mathrm{ReLu}(\mu_i(X_l^2) - (\mu_i(X_l))^2) + \epsilon} \tag{4}$$

In equations 1, 2, 3, 4 u_i refers to the mean value for node i, X^l is the node feature for the layer l, d_i is the number of neighbors for node i, which remains unchanged in the case of a static graph and N_i to the set of neighbors for the node i. σ_i refers to the standard deviation value for node i, and X^l represents the node feature for layer l. The ReLu activation function filters out the negative values, and a small value is added so that the function stays differentiable.

When the degree of the node is high, four aggregators might not be enough to extract useful information from the neighboring nodes. In that case, PNA uses high moments to aggregate information.

$$M_n(X) = \sqrt[n]{E[X-u]^n} \tag{5}$$

In some cases where two nodes have similar neighborhood feature vectors but different node degrees, the aggregators discussed above need to distinguish between two such nodes. To solve such a problem, PNA uses the logarithm of degree-based scalars as represented in equations 6, 7.

$$S_{amp}(d) = \frac{\log(d+1)}{\lambda} \tag{6}$$

$$\lambda = \frac{1}{|train|} \sum_{i \to train} \log(d_i + 1) \tag{7}$$

Here λ is the normalization parameter, and d is the degree of nodes. The degree-based scalar is further generalized by adding the variable α, which is negative for attenuation, positive for amplification, and zero for no scaling. The final expression for degree-based scalar is given in the equation 8

$$S(d, a) = \left(\frac{\log(d+1)}{\lambda}\right)^{\alpha} \tag{8}$$

Here, the value of α varies from -1 to +1.

The three types of degree scalar and four aggregators are combined, resulting in 12 message vectors from each neighboring node. The expression below explains the message passing and updation process for a layer of gnn using the PNA framework.

$$X_i^{t+1} = U\left(X_i^t, \oplus_{(i,j) \in E} M(X_i^t, E_{j \to i}, X_j^t)\right) \tag{9}$$

The $E_{j \to i}$ is the feature of the edge connecting node j to i. M and U are neural networks responsible for the aggregation and updation process.

B. GRU

In this work, we use GRU to capture the temporal dependence. It takes the current input and hidden representation from the past data to determine the current representation. GRU can model sequential data, capture long-term dependencies and handle the vanishing gradient problem. By incorporating the GRU alongside other components, such as PNA [10] and attention layers, the model can effectively leverage the temporal dynamics and dependencies in the attack data.

GRU achieves this through the use of gating mechanism, which control the flow of information within the network. It consists of two main gates: the update and reset gates. The update gate decides to what extent the previous hidden state should be used to update the new state, while the reset gate decides to what extent past information should be forgotten.

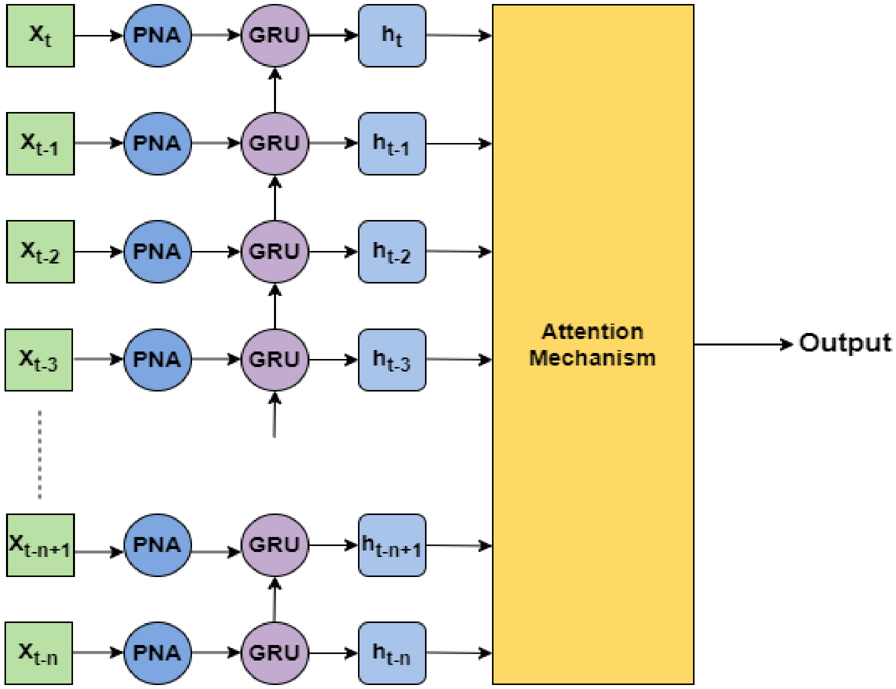

Fig. 1. **The Proposed PANGA Model Architecture** 1.Cyber attack data in the form of a graph at each time step is passed through the PNA layer where the embeddings of the graph nodes are updated by considering spatial correlation among neighboring nodes. 2.The GRU layer takes past time hidden state and current graph node embedding as input to determine the hidden state for the current time step. The attention mechanism generates random weights from a uniform distribution to determine the context vector.

These gates are computed using specific equations: the update gate equation is

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{10}$$

, and the reset gate equation is

$$r_t = \sigma(W_r x_t + U_r h_{t1} + b_r) \tag{11}$$

These equations use input $\mathbf{x_t}$ at time step t and the previous hidden state $\mathbf{h_{t-1}}$, combined with learnable weight matrices $\mathbf{W_z}$, $\mathbf{W_r}$, $\mathbf{U}_z$, $\mathbf{U_r}$, and bias vectors $\mathbf{b_z}$, $\mathbf{b_r}$. The gates regulate the amount of information that is updated and forgotten, allowing GRU to selectively retain important information and effectively capture long-term dependencies in the data.

Then we determine the new hidden state candidate based on the previous hidden state, the input and the reset gate. It is calculated using the equation

$$\tilde{h}_t = tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{12}$$

The hidden state update process combines the information from the candidate activation and the previous hidden state, controlled by the update gate. It is given by

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{13}$$

C. Attention model

The attention model improves model performance by selectively focusing on relevant parts of the input sequence, allowing the model to emphasize informative features and ignore other parts. The attention mechanism helps in the modeling of long-term dependencies, capturing complex relationships spanning over multiple time steps. Additionally, the interpretability of attention weights provides insights into the model's decision-making process, enabling a better understanding of the correlation between input and output sequences.

Here, we implement a soft attention model above the GRU layer. GRU caters to handling the temporal dependencies of the model. The attention mechanism assigns weight to each past hidden vector determined by the GRU layer till a certain time. The assigned weight to each hidden layer in the past is proportional to their relevance to the current input vector.

The PNA model acts as an encoder and provides an embedding vector to each node of the graph for each time unit, depending on the interaction between different nodes of the graph. The embedded vector of the graph for each time unit will pass through the GRU model and the hidden representation of nodes is obtained in $H = [h_1, h_2, h_3, ...h_t]$ form, where h_t represent the embeddings of each node of the graph at time t. The soft attention model has a scoring function that determines the weight of each hidden state. The scoring function is implemented using various approaches, such as a dot product, a multilayer neural network, or a bilinear function. In this work, random weights are generated using uniform distribution and then each weight is normalized using the softmax function to determine the attention weights of each hidden state, The context vector is determined by summing the product of attention and hidden state of each time unit.

$$e = U(0,1) \tag{14}$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_{j=1}^{T} \exp(e_j)} \tag{15}$$

$$c_t = \sum_{i=1}^{T} \alpha_i h_i \tag{16}$$

Here, e refers to the random weight vector generated using a uniform distribution with minimum and maximum values as 0, 1. The size of the vector is the same as the context window size. α_i is the attention weight determined by applying the softmax function to e_i and c_t refers to the context vector at time unit t.

D. The Proposed Forecasting Model: PANGA

In order to forecast the number of cyber attacks in a network, we propose a model PANGA as described in figure 1. It is a multi-layered neural network architecture that incorporates PNA, a GRU model, and a soft attention mechanism. The PNA model handles the spatial dependence of the dataset. It acts as an encoder and updates the embedding of the graph node by interacting with the node's and its neighbor's embedding. PNA algorithms take the frequency of transactions and packet size into account to determine the extent of dependence between two neighboring nodes. Above the PNA layer, we have the GRU model. GRU model takes care of the temporal embedding of the dataset. It enables the model to capture sequential dependencies and temporal patterns in the data. It can learn to encode information from past time steps and use it to make predictions. GRU is sensitive to the order of the input sequence, which might affect its ability to handle long-term dependency and focus on relevant information.

V. Dataset

In this work, we use CIDDS-001 (Coburg Intrusion Detection Data Set) [11], which is a labeled flow-based dataset. It has been specifically created for the purpose of evaluating anomaly-based intrusion detection systems. The dataset has a unidirectional netflow. It includes traffic data collected from two servers: the OpenStack server and the external server. Our dataset had 172839 instances of traffic sampled from both the traffic data collected from servers. The dataset has 16 attributes, including the source IP address and destination IP address feature. We have 10539 and 10478 unique source IP addresses and destination IP addresses.

A. Data Preprocessing

The CIDDS-001 dataset has 172839 rows and 16 columns. After converting the time stamp column to pandas date-time format, we sorted the dataset using the time stamp column in ascending order. We removed unnecessary columns. Our target column had three unique values: normal, malicious, and unknown; we replaced normal and unknown with 0 and malicious with 1 as we were only interested in forecasting malicious flows between two IP addresses. Categorical columns had high cardinality, so we applied target encoding [26] to convert categorical data to numbers. We used StandardScaler to standardize the dataset.

B. Graph Building

We built the static graph for the forecasting task for the networked device. The edges between two nodes of the graph and their weights were static, but the node values were dynamic and changed with time as described in figure 2. While working on a short-term prediction problem, the spatial dependencies between the nodes of the graph would not change much, and in a dynamic graph where new nodes are getting added/removed, and new edges are getting formed/removed every 15, 20, or 30 minutes, this would have resulted in the loss of information, as a result, the static graph is build.
For forecasting cyber attacks 15 minutes, 20 minutes, and 30 minutes ahead, we prepare three separate node value datasets and one standard adjacency matrix. We aggregated the dataset by 15, 20, and 30-minute periods and added the aggregated cyber attack number. The node value matrix had 349, 523, and 698 rows for 15, 20, and 30-minute periods. We had 10539, and 10478 unique source IP addresses and destination IP addresses, respectively; we took the union of those values and formed a two-dimensional matrix of shape 10539 by 10539. The adjacency matrix is a weighted unsymmetric matrix; if the IP address placed at the row had sent any packet of data to the IP address present in row j, then mat[i,j] will have a non-zero value. The weight of the edge between two nodes i and j is determined by the average package size sent from node i to j and the number of nodes i has sent data to node j.

$$W_{ij} = \frac{Data\ sent\ by\ i\ to\ j}{n_{ij}} + n_{ij} \tag{17}$$

Here, W_{ij} is the weight of the edge from node i to j, and n represents the number of times the data has been sent to node j from node i.

VI. Results

A. Evaluation Metric

We used Mean Square Error(MSE) and Coefficient of Determination(r^2) to evaluate the performance of our proposed model. MSE determines the mean of the squared difference between the predicted values and the ground truth in the dataset. The closer the predicted value is to the true value, the lower the MSE value. The coefficient of determination(r^2) given by equation 19 explains the proportion of the variance in the target variable that can be explained by the independent variables in a regression model. It shows how well the regression model fits the data that was observed.

$$\text{MSE} = \frac{1}{tn} \sum_{j=1}^{t} \sum_{i=1}^{n} (y_{ij} - \hat{y}_{ij})^2 \tag{18}$$

$$r^2 = 1 - \frac{\sum_{j=1}^{t} \sum_{i=1}^{n} (y_{ij} - \hat{y}_{ij})^2}{\sum_{j=1}^{t} \sum_{i=1}^{n} (y_{ij} - \bar{y})^2} \tag{19}$$

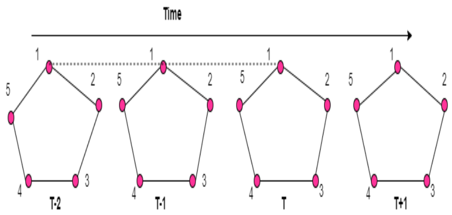

Fig. 2. **Static graph with dynamic node features** The graph node 1 at time T is spatially correlated with its neighbors 2 and 5 and temporally correlated with node 1 of the graph at time T-2, T-1 and T+1.

TABLE I
Comparison Table for 15, 20 and 30 minute forecasting task

	Proposed PANGA		A3Tgcn		Tgcn	
Time Period	MSE	r	MSE	r	MSE	r
15	.09	.78	.09	.79	.85	.54
20	0.040	.83	.068	.81	.66	.6
30	0.027	.87	.052	0.83	.49	.64

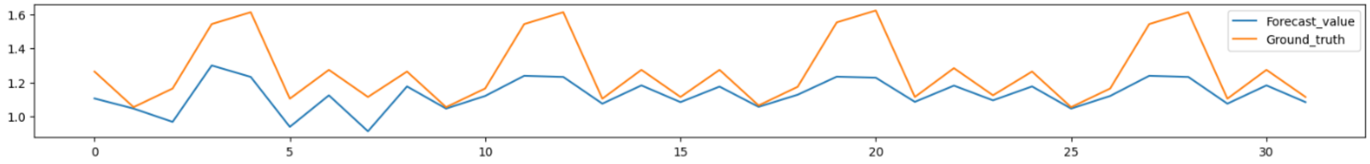

Fig. 3. Visual depiction of 30-minute forecasting task

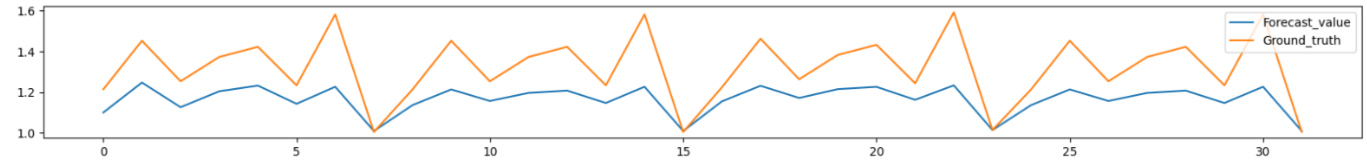

Fig. 4. Visual depiction of 20-minute forecasting task

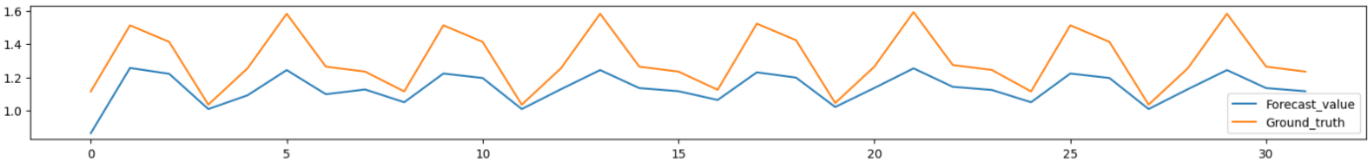

Fig. 5. Visual depiction of 15-minute forecasting task

Here y_{ij} and $\hat{y_{ij}}$ are real and predicted values for node i at time j, $\bar{y}$ indicates the mean value of ground truth. n represents the total number of nodes present in the network and t is the output time unit.

B. Hyperparameter Setting

We experimented with different values for hyperparameters before settling on the ones that gave the most optimal result. Epochs, learning rate, hidden layer size, and batch size are taken as hyperparameters. We train our model for 50 epochs with a learning rate of .002. The model has one hidden layer with a size of 32 and a batch size of 8. We use Adam optimizer and used regression loss as the loss function. 85% data is used for model training and the rest 15% for model performance evaluation. The context sizes finalized for the attention mechanism were taken as 12.

C. Analysis

Table 1 compares our model performance against two other deep learning models built for traffic forecasting tasks. A3TGCN[22] and TGCN[27] were initially trained on the taxi trajectory dataset in Shenzhen City and the loop detector dataset in Los Angeles to forecast traffic speed. The building blocks of the A3TGCN model are GCN, GRU, and attention layer, while our architecture consists of PNA, GRU and attention layer. We compared our model performance against

TABLE II
PERTURBATION ANALYSIS

Standard deviation	unperturbed data	0.1	0.2	0.25	0.5	0.75	1
MSE	0.027	.060	.095	.11	.29	.62	1.12
R^2	.87	.834	.81	.79	.76	.7	.62

A3Tgcn model on CIDDS-001[11] dataset as it elucidates the differences in how PNA and the GCN components of the model handle spatial dependence among neighboring nodes and their capabilities in developing the embedding for the graph nodes. The TGCN does not have attention layer, so comparing its performance against A3TGCN and PANGA which incorporate attention mechanisms, can provide valuable insights into the importance of attention mechanisms for cyber attack forecasting task. We trained these two models on CIDDS-001[11] dataset and compared them with our model performance. The TGCN performed considerably worse than the other two models. For the 15-minute of forecasting task, the MSE value for TGCN is close to 1, while the MSE value of the other two models is one-eighth of the MSE of TGCN; the coefficient of determination is also 66% higher for attention-based models compared to TGCN. For the 20 and 30-minute forecasting tasks, the performance of the TGCN model is considerably better than 15 minutes of forecasting as its MSE value is down by 20% and 40% for 20 minutes and 30-minute tasks. However, it is still nowhere as accurate as the other two attention-based models. This shows that the attention mechanism is helpful in malware forecasting tasks by focusing on the relevant parts of the input sequence and capturing dependencies between those parts of the input sequence.

For 15 minutes forecasting task, both A3Tgcn and PANGA performed almost the same, but for the 20 and 30-minute forecasting task, the PANGA model performed better than A3TGCN by a margin. For 20 minutes forecasting task, although the coefficient of the determination value is not much different, PANGA has 40% less MSE value compared to the A3TGCN model. Again PANGA model has a 40% lesser MSE value and a 5% high coefficient of determination value compared to the A3TGCN model for 30 minutes forecasting task. The spatial dependency in the PANGA model is handled by the PNA model, which is considerably better at developing embedding than GCN as it utilizes four permutation invariant methods along with 3-degree scalars for the message passing process compared to the GCN model, which only uses mean to pass information among nodes in the graph.

The performance of all three models worsens as we reduce the time step size. They perform best for 30-minute steps and considerably worse for 15-minute steps. The possible reason behind these observations is that 15 minutes time step could be too small to capture any pattern, and the noise-to-pattern ratio could be too high to give us any meaningful result. At the same time, as we increase the time step size, the model is able to capture relevant information and pattern to carry out the forecasting task.

D. Perturbation analysis

We added random noise sourced from Gaussian distribution with zero mean and standard distribution taken from this set $[0.1, 0.2, 0.25, 0.5, 0.75, 1]$ to test the robustness of our model. As is evident from Table II, the performance of the model gradually deteriorated as we increased the standard deviation of the noise. Till $\sigma \leq .25$, the MSE was 0.11, and the coefficient of determination has also not declined below 0.8, this shows that model can successfully withstand any adversary attack with $\sigma \leq 0.25$.

VII. CONCLUSION

In this paper, we propose a model, PANGA to forecast the cyber attack using the short-term time series forecasting method. We combine PNA, GRU, and attention mechanisms to forecast the number of cyber attacks for 20, 15, and 30 minutes. We converted the tabular data to graphical data and employed PNA to handle the spatial dependencies while the GRU layer dealt with the dynamic dependencies. The attention layer focused on relevant parts of the input sentence to determine the context vectors. The model was evaluated on the CIDDS-001 dataset. Our model achieved an MSE value of 0.027 and a coefficient of determination of .87 for a 30-minute forecasting task.

We have developed a short-term forecasting model for cyber attacks. As part of our future work, we plan to create a custom dataset for a long-term forecasting task. In our current model, only node features are considered dynamic, while the rest of the graph remains static. For the long-term prediction task, we intend to extend the model to incorporate dynamic changes in both the nodes and the edges between them, as well as the node features.

REFERENCES

[1] T. wire staff, *5 AIIMS Servers Hacked, 1.3 TB Data Encrypted in Recent Cyberattack, Govt Tells RS*, https://thewire.in/government/aiims-servers-cyberattack-ransomware-rajya-sabha, [Online; accessed 11-March-2023], 2022.

[2] World Economic Forum, "The global risks report 2022 17th edition," 2022.

[3] C. Brooks, *Cybersecurity Trends Statistics; More Sophisticated And Persistent Threats So Far In 2023*, https://www.forbes.com/sites/chuckbrooks/2023/05/05/cybersecurity-trends--statistics-more-sophisticated-and-persistent-threats-so-far-in-2023/?sh=7d42181b7cb6, [Online; accessed 5-june-2023], 2023.

[4] Statistica, *Estimate cost from cybersecurity Worldwide 2017-2028*, https://www.statista.com/statistics/1280009/cost-cybercrime-worldwide/, [Online; accessed 5-june-2023], 2023.

[5] S. B.S., N. S., N. Kashyap, and S. D.N., "Providing cyber security using artificial intelligence – a survey," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 2019, pp. 717–720. DOI: 10.1109/ICCMC.2019.8819719.

[6] A. Shenfield, D. Day, and A. Ayesh, "Intelligent intrusion detection systems using artificial neural networks," *ICT Express*, vol. 4, no. 2, pp. 95–99, 2018, Artificial Intelligence and Machine Learning Approaches to Communication, ISSN: 2405-9595.

[7] D. University, *ARIMA models for time series forecasting*, https://people.duke.edu/~rnau/411arim.htm, [Online; accessed 5-june-2023], 2017.

[8] penn state eberly college of science, *Vector Autoregressive models VAR(p) models*, https://online.stat.psu.edu/stat510/lesson/11/11.2, [Online; accessed 5-june-2023], 2017.

[9] wikipedia, *Bayesian structural time series*, https://en.wikipedia.org/wiki/Bayesian_structural_time_series, [Online; accessed 5-june-2023], 2017.

[10] G. Corso, L. Cavalleri, D. Beaini, P. Liò, and P. Veličković, "Principal neighbourhood aggregation for graph nets," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 260–13 271, 2020.

[11] https://github.com/markusring/CIDDS, *CIDDS-001*, https://github.com/markusring/CIDDS, [Online; accessed 5-March-2023], 2017.

[12] G. Werner, S. Yang, and K. McConky, "Time series forecasting of cyber attack intensity," in *Proceedings of the 12th Annual Conference on cyber and information security research*, 2017, pp. 1–3.

[13] https://www.hackmageddon.com/, *HACKMAGEDDON*, https://www.hackmageddon.com/, [Online; accessed 11-June-2023], 2023.

[14] F. Quinkert, T. Holz, K. Hossain, E. Ferrara, and K. Lerman, "Raptor: Ransomware attack predictor," *arXiv preprint arXiv:1803.01598*, 2018.

[15] J. Z. Bakdash, S. Hutchinson, E. G. Zaroukian, *et al.*, "Malware in the future? forecasting of analyst detection of cyber events," *Journal of Cybersecurity*, vol. 4, no. 1, tyy007, 2018.

[16] J. Hui, *Hidden Markov Model*, https://jonathan-hui.medium.com/machine-learning-hidden-markov-model-hmm-31660d217a61, [Online; accessed 11-Febuary-2023], 2019.

[17] S. Maitra, *Prediction using Bayesian State Space Model*, https://towardsdatascience.com/natural-gas-price-prediction-using-bayesian-state-space-model-for-time-series-forecasting-f630dda1c808, [Online; accessed 11- Febuary-2023], 2020.

[18] X. Fang, M. Xu, S. Xu, and P. Zhao, "A deep learning framework for predicting cyber attacks rates," *EURASIP Journal on Information security*, vol. 2019, pp. 1–11, 2019.

[19] K. Albulayhi and Q. A. Al-Haija, "Early-stage malware and ransomware forecasting in the short-term future using regression-based neural network technique," in *2022 14th International Conference on Computational Intelligence and Communication Networks (CICN)*, IEEE, 2022, pp. 735–742.

[20] Z. Almahmoud, P. D. Yoo, O. Alhussein, I. Farhat, and E. Damiani, "A holistic and proactive approach to forecasting cyber threats," *Scientific Reports*, vol. 13, no. 1, p. 8049, 2023.

[21] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 1234–1241, Apr. 2020. DOI: 10.1609/aaai.v34i01.5477. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/5477.

[22] J. Bai, J. Zhu, Y. Song, *et al.*, "A3t-gcn: Attention temporal graph convolutional network for traffic forecasting," *ISPRS International Journal of Geo-Information*, vol. 10, no. 7, 2021, ISSN: 2220-9964. DOI: 10.3390/ijgi10070485. [Online]. Available: https://www.mdpi.com/2220-9964/10/7/485.

[23] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence Organization, Jul. 2018. DOI: 10.24963/ijcai.2018/505.

[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016. arXiv: 1609.02907. [Online]. Available: http://arxiv.org/abs/1609.02907.

[25] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, *Graph attention networks*, 2018. arXiv: 1710.10903 [stat.ML].

[26] H20.ai, *Target Encoding*, https://towardsdatascience.com/dealing-with-categorical-variables-by-using-target-encoder-a0f1733a4c69//, [Online; accessed 11-Febuary-2023], 2021.

[27] L. Zhao, Y. Song, C. Zhang, *et al.*, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020. DOI: 10.1109/TITS.2019.2935152.