International Journal of Research in Engineering, Science and Management
Volume 8, Issue 11, November 2025
www.ijresm.com | E-ISSN: 2581-5792 | RESAIM Publishing (www.resaim.com)

Using Gated Recurrent Unit Recurrent Neural Network for Detection of HTTP Flood Attack

Ezekiel Cheruiyot Ronoh*
School of Technology, KCA University, Nairobi, Kenya

Abstract*: The HTTP flood attacks has been on the rise in disruption of digital services and infrastructure by cybercriminals using readily available DDoS execution tools on the internet. These attacks disrupt online services by bombarding web servers with HTTP Requests to deny legitimate users access to online services. Prevention of these attacks require proactive defense mechanism to differentiate malicious HTTP Requests from legitimate users HTTP Requests in real time, the increased number of online users require more resources for monitoring. Therefore, early detection of the attacks can be utilized to implement reactive mechanism to reduce the effect of attacks by preventing escalation and reduce losses incurred by the online businesses. This research paper presents the use of GRU model to predict Web Server access logs patterns for early detection of the HTTP Flood Attacks. The hidden state in GRU provides memory of past events to enable accurate prediction of attacks based on past trends of HTTP Requests frequency in the Web Server. The model was trained on Web Server Access Logs dataset through multivariate regression analysis and prediction of HTTP Request frequency. This involved analysis of 10,365,093 HTTP Requests from 'WEB SERVER ACCESS LOG' dataset and 47,742 HTTP Requests from 'EPA-HTTP' dataset. The GRU model prediction was able to achieve the Mean Absolute Error of 0.0188 and 0.0149 respectively indicating high prediction accuracy. Further comparison with LSTM Model using the same hyperparameters and the two datasets, indicated slightly high accuracy of GRU Model of 0.0188 against 0.019 for LSTM Model for the Web Server Access Log dataset and 0.0149 for the GRU model against 0.0158 for the LSTM model using the EPA-HTTP dataset. This research shows that the Gated Recurrent Unit (GRU) simplified modern architecture of LSTM Model can be deployed to detect HTTP Flood Attacks fast by predicting frequency of HTTP Requests Methods received by the Web Server. The highspeed performance and increased accuracy of GRU will enable the analysis of huge the access logs with efficient utilization of resources for real time detection of attacks when they occur.

***Keywords*: HTTP(S), DDOS, GRU, LSTM.**

1. Introduction

Online services providers have leveraged the wide adoption of internet to gain competitive advantage by providing variety of goods and services to the customers cheaply and efficiently (Sharma, 2024). HTTP Flood Attacks have become a major threat to availability of online services causing disruption which leads to financial losses to online businesses and service providers. These attacks leverage the use of emerging technologies like Artificial Intelligence to increase their effectiveness in terms of speed of execution and impact, presenting a huge challenge in detection and prevention of these attacks.

This research focused on use of Gated Recurrent Unit (GRU) Neural Network model to predict HTTP Methods frequency in a Web Server for quick detection of attacks when they occur. The early detection of attacks can be used to trigger a reactive measure to counter the attack and reduce the impact of HTTP Flood Attacks and prevents online service disruption which ensure continuous availability of the online services.

GRU model is effective in detection of HTTP Flood Attacks by learning patterns of features from network traffic (Mittal, 2023). The high performance of GRU (Ireri, 2020), which is a type of Recurrent Neural Network (RNN), is achieved by use of gating mechanisms to selectively update hidden state of the model. The parallel execution of Artificial Neural Network (ANN) structure for parallel information processing (Xu, 2019) can process multi-channel real-time series data reducing the time taken in processing of data.

Web server Access Logs can be analyzed in real time by use of Regular Expression which enable faster extraction of parameters in huge datasets to produce time series datasets. This dataset is utilized in the prediction of HTTP Methods frequency patterns in the Web Server, which can be utilized to prevent attacks which deviate from the patterns predicted.

To enable faster detection with high number of online users accessing services online, Attention Mechanism, which is a technique in machine learning that utilizes selective focus on the salient parts of data in order to increase efficiency in the understanding of the data, is deployed to reduce the number of features. This increases the accuracy and robustness of forecasting (Wang, 2023).

This research paper proposes the use GRU model for fast, cost-effective and cost-efficient detection of HTTP Flood Attack preventing disruption of online services using HTTP Methods frequency as a selected parameter in the Web Server access logs.

A. Scope

The HTTP Flood attacks can be deployed by malicious actors by bombarding the Web Server using any of the nine HTTP Methods, hence there is need for implementation of a detection mechanism which can detect attacks from all the vectors. The

*Corresponding author: 2300827@students.kcau.ac.ke

current body of knowledge on the detection of HTTP Flood Attack does not present an integrated, fast, cost effective and cost efficient detection mechanism, for instance Reza, et al. (Reza, 2022) deployed Machine Learning to detect only attacks from GET and POST attack vectors, which does not prevent the nine potential vectors represented the HTTP Flood Method: GET, HEAD, CONNECT, OPTIONS, TRACE, PATCH, PUT, DELETE and POST.

Application layer attacks are more destructive due to the capability to exhaust server and network resources faster. The research will focus on detection of Application Layer attack specifically HTTP Flood Attack to prevent the depletion of server resources. The research paper uses GRU model to detect HTTP Flood Attacks that can cause depletion of server resources preventing normal users from accessing online services. The webserver access logs will provide the data for formulation of the model.

B. Objective

The aim of the research is to Use Gated Recurrent Unit (GRU) Recurrent Neural Network model in prediction future patterns of the HTTP Method Requests frequency on a web server. The GRU model has been proven to be fast and efficient in non-complex sequence dataset (Cahuantzi, 2024). Regular Expression is used to extract the HTTP Methods frequencies from the web server access logs, to form a time series dataset. GRU model aims to provide a means of fast detection of attacks by:

a) Provide real time defense of HTTP Flood Attacks by utilization of server access logs to predict HTTP flood requests.
b) Provide low cost and efficient mechanism for the detection of HTTP Flood Attacks.
c) Utilize parallel execution of Artificial Neural Networks to enable the detection HTTP Flood Attack in a short time to prevent escalation of attacks.

This will provide a cost-effective mitigation of HTTP Flood Attacks to reduce the expenditure of resources when online services are disrupted.

2. Literature Review

There has been tremendous effort to detect HTTP Flood attacks using Deep Learning models, of note is LSTM model which have become the prominent Recurrent Neural Network architectures that have been deployed in forecasting of time series data.

Aswad, et al. (Aswad, 2022) utilized deep learning to manage DDos attacks using four algorithms; Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), convolutional Neural Network and CNN-bidirectional LSTM (CNN-BiLSTM) and compared performance using confusion matrix. This research used CICIDS2017 dataset for training and testing of the models. The results show that the CNN-BiLSTM hybrid model achieved an accuracy of 99.76%, LSTM 99.76%, RNN 99.58%, and CNN 99.82%. However, the research did not consider the need to cater for huge data generated in real time and the use of GRU for detection of DDoS attacks.

Salinas, et al. (Salinas, 2019), proposed application of deep learning technique DeepAR which is LSTM-based recurrent neural network, for forecasting to overcome challenges faced by classical approaches. DeepAR was tested against Matrix Factorization technique using Parts dataset containing monthly sales time series data, Electricity dataset containing electricity consumption hourly time series data and Traffic dataset containing hourly car lanes occupancy rate. The DeepAR outperformed Matrix Factorization having 1.00 RMSE against 1.15 in the Electricity dataset and 0.42 RMSE against 0.43 in the Traffic dataset. Demonstrated the learning jointly from multiple time series real world problems and providing better accuracy than the classical methods. Multiple time series problems have different magnitudes and the distribution of these magnitudes is strongly skewed. The accuracy of the deep learning model improves by around 15%. However, the development of GRU model which users less parameters than LSTM presents a better option for faster processing to mitigate current challenges of huge volume of data generated.

Feature selection has been deployed to increase the model accuracy and performance by Azmi, et al. (Bansode, 2024) and Raza, et al. (Raza, 2024), therefore the selection of HTTP Method frequency for prediction of attacks has been chosen as the prominent feature for detection of attacks on a web server.

The development of the Gated Recurrent Unit (GRU) presents a faster model for prediction of time series datasets, this research utilized GRU model which is a simplification of the LSTM architecture that enable high performance and efficiency when used in low complex dataset compared to LSTM (Kaushik, 2025). The studies by Shiri, et al. (Shiri, 2024) and Cahuantzi, et al. (Cahuantzi, 2024) have compared the performance of deep learning models RNN, LSTM, CNN and GRU on accuracy, time of execution and complexity, with conclusion that the GRU model performs faster, has high accuracy of detection and less execution data on less complex datasets.

From the analysis of the solutions that have been proposed by the studies evaluated above, the gap identified is the implementation of faster detection of HTTP Flood Attack using the GRU architecture which is a fast, high performance, efficient and cost-effective mechanism for detecting attacks. Early detection enables deployment of a reactive mechanism to counter the attacks, reducing loss of resources.

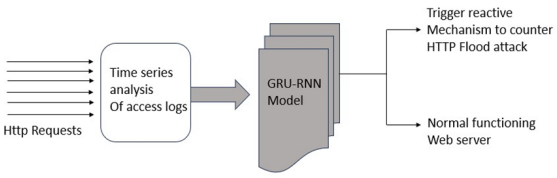

Fig. 1. Architecture

3. Methodology

This research undertook quantitative research that analyzed the causal relationship between the frequency of HTTP Requests Methods variable and the depletion of Web Server

resources. The proposed use of the GRU model for detection of HTTP Flood attacks involved several stages outlined below:

A. Data Collection

The datasets utilized in the research are:

a) Web Server Access Log: Web Server Access Dataset for Model Training and Validation used secondary dataset from Kaggle.
b) EPA HTTP Dataset for Model Testing.

B. Data Preprocessing

1) Feature Extraction

Regular Expressions are able to quickly parse through large amount of data accurately looking for matches of patterns which makes them invaluable when it comes time-sensitive searches (Lenovo, 2024). The project used the following python script to extract HTTP Methods frequency in time period of 10 seconds:

```
# EXTRACT THE HTTP METHODS FREQUENCY FROM THE ACCESS LOG FILE.
with open('C:/Users/hp/Desktop/archive/access.log') as log:
    method_frequency = defaultdict(int)
    window_frequency = defaultdict(lambda: defaultdict(int))
    window_sums = defaultdict(int
    for line in log.readlines():
        match = re.match(r'(.+) - - \[(\d{2}\/\w{3}\/\d{4}:\d{2}:\d{2}:\d{2}) \+\d{4}\] \"(GET|CONNECT|HEAD|OPTIONS|TRACE|PATCH|PUT| DELETE|POST) .+', line)
        if match:
            timestamp = datetime.strptime(match.group(2), '%d/%b/%Y:%H:%M:%S')
            method = match.group(3).split()[0]
            method_frequency[method] += 1
            window_start = timestamp - timedelta(seconds=timestamp.second % 10)
            window_frequency[window_start][method] += 1
```

2) Data Cleaning

Enabled accurate insights are generated from the data, it involved handling of missing values and Missing values imputation strategy to ensure accuracy and reliability.

3) Data Transformation

Involved data transformation, splitting the datasets into training and testing datasets, create sequences and convert the datasets to Pytorch Tensors.

C. Customization and Initialization of Gated Recurrent Unit (GRU) Neural Network and Long Short-Term Memory (LSTM) Neural Network

The project utilized the PyTorch Framework to implement the GRU and LSTM models.

1) GRU Model

```
class MultivariateGRU(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers, output_size):
        super(MultivariateGRU, self).__init__()
        self.gru = nn.GRU(input_size, hidden_size, num_layers, batch_first=True)
        self.fc = nn.Linear(hidden_size, output_size)
        # Dropout layer
        self.dropout = nn.Dropout(dropout_rate)
    def forward(self, x):
        out, _ = self.gru(x)
        out = self.fc(out[:, -1, :])
        # Apply dropout
        out = self.dropout(out)
        # Fully connected layer
        out = self.fc(out)
        return out
******
```

2) LSTM Model

```
*****
class MultivariateLSTM(nn.Module):
    def __init__(self, input_size, hidden_size, num_layers, output_size):
        super(MultivariateLSTM, self).__init__()
        self.lstm = nn.LSTM(input_size, hidden_size, num_layers, batch_first=True)
        self.fc = nn.Linear(hidden_size, output_size)
        # Dropout layer
        self.dropout = nn.Dropout(dropout_rate)
    def forward(self, x):
        out, _ = self.lstm(x)
        out = self.fc(out[:, -1, :])
        # Apply dropout
        out = self.dropout(out)
        # Fully connected layer
        out = self.fc(out)
        return out
```

D. Model Training, Evaluation and Testing

This enabled the model to generate the relevant parameter that can be utilized in the future prediction by the model

E. Model Prediction

The model was used for forecasting of the future patterns of HTTP Methods frequency.

4. Results

A. Web Server Access Log Dataset

Table 1
Extracted HTTP methods frequency

No.	HTTP Method	Total No. of Requests
1	GET	10190003
2	POST	139155
3	HEAD	34501
4	OPTIONS	1424
5	CONNECT	10
6	PUT	0
7	DELETE	0
8	TRACE	0
9	CONNECT	0

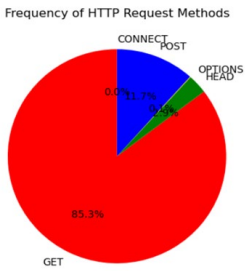

Fig. 2. Extracted HTTP methods frequency Pie-plot

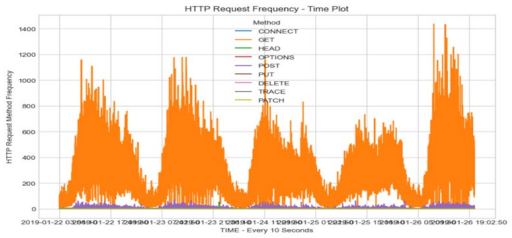

Fig. 3. Extracted HTTP methods frequency time plot

B. *EPA HTTP Dataset*

Table 2
Extracted HTTP methods frequency

No.	HTTP Method	Total No. of Requests
1	GET	10190003
2	POST	139155
3	HEAD	34501
4	OPTIONS	1424
5	CONNECT	10
6	PUT	0
7	DELETE	0
8	TRACE	0
9	CONNECT	0

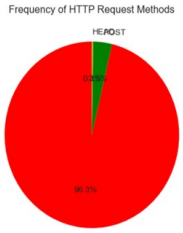

Fig. 4. Extracted HTTP methods frequency Pie-plot

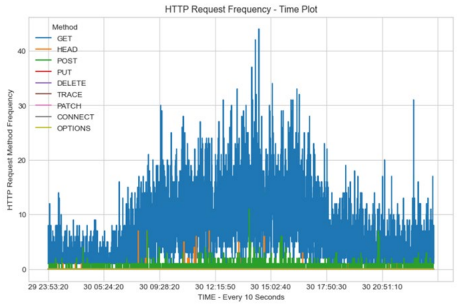

Fig. 5. Extracted HTTP methods frequency time plot

1) *Initialization of the GRU and LSTM Models*

The models were implemented using the PyTorch Framework and the following values were initialized to train, validate and test for prediction of HTTP Methods frequency.

Table 3
Models hyperparameters

No.\Item	Hyperparameter	Value
1	Epoch	100
2	Number of Layers	5
3	Input layer	9
4	Hidden layers	3
5	Output layer	9
6	Sequence Length	120
7	Optimizer	Adam
8	Learning rate	0.01
9	Loss Function	Mean Absolute Error
10	Drop Out Rate	0.2

C. *Training and Testing of the Models*

Fig. 6. Training of the GRU model using the web server access log dataset

Fig. 7. Testing the GRU Model using EPA – HTTP Dataset

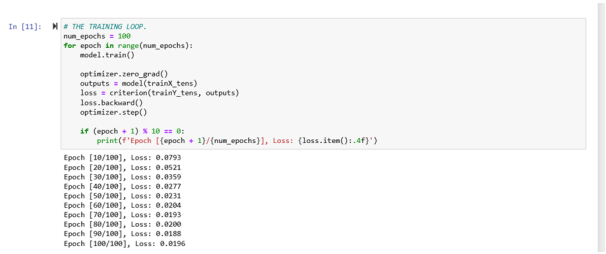

Fig. 8. Training of the LSTM Model using web server access log dataset

Fig. 9. Testing the LSTM model using EPA-HTTP dataset

Table 4
Comparison between the GRU and LSTM models on the mean absolute error metric

Dataset	Gru MAE Metric	LSTM MAE Metric
Web Server Access Log	0.0188	0.0190
EPA-HTTP	0.0149	0.0158

5. Conclusion

There were high performance and accuracy of GRU model compared with LSTM Model and also the duration taken to train the two models differed significantly with GRU Model

taking less time than LSTM to complete one round of 100 epochs for the two datasets.

The GRU model can successfully be utilized in the detection of HTTP Flood Attacks due to its capability to utilized few parameters to train, hence efficient. Also, parallel ingestion of features enables the processing of many features simultaneously leading to less resource consumption. This is applicable today when there is an increase of online users and increased hardware capacity enabling a single web server to host services that can be accessed by huge number of users hence the need for detection of attacks by analysis of this huge data.

6. Recommendation for Future Research

The parallel processing feature of Artificial Neural Networks (Xu, 2019), provides an efficient mechanism that can be used in profiling of individual users that are accessing the web services simultaneously. This can enable the prediction of their behavior using user specific features that were implemented in this research. This has the potential of implementing mitigation of DDoS Attacks by early detection of malicious users and selective termination of the malicious user when they are identified.

References

[1] S. P. Bansode and R. Patil, "A hybrid feature selection approach incorporating mutual information and genetic algorithm for web server attack detection," *Indian J. Sci. Technol.*, vol. 17, no. 14, pp. 325–332, 2024.

[2] E. Cahuantzi et al., "A comparison of LSTM and GRU networks for time series prediction," *Tech. Rep.*, Univ. of Manchester, Manchester, U.K., Jan. 2024, pp. 1–15.

[3] K. A. Ireri and co-authors, "Towards optimization of the gated recurrent unit (GRU) for regression modeling," *Int. J. Social Sci. Inf. Technol.*, vol. 6, no. 10, pp. 1–10, Oct. 2020.

[4] R. Kaushik, "LSTM vs GRU vs transformers: Choosing the right model for your data," *LinkedIn Article*, Feb. 15, 2025. [Online]. Available: https://www.linkedin.com/pulse/lstm-vs-gru-transformers-choosing-right-model-your-data-rohan-kaushik-wwkyc/

[5] Lenovo, "What is regex?," *Lenovo Glossary*, Jul. 1, 2024. [Online]. Available: https://www.lenovo.com/us/en/glossary/regex/

[6] A. Mittal et al., "DDoS attacks detection using a deep neural network model," in *Adv. Comput.*, 2023, pp. 169–182.

[7] M. S. Raza, M. N. A. Sheikh, I. S. Hwang, and M. S. Ab-Rahman, "Feature-selection-based DDoS attack detection using AI algorithms," *Telecom*, vol. 5, no. 2, pp. 333–346, Apr. 2024.

[8] R. Mohammadi, C. Lal, and M. Conti, "HTTPScout: A machine learning based countermeasure for HTTP flood attacks in SDN," *Int. J. Inf. Secur.*, vol. 21, pp. 367–379, 2022.

[9] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, "DeepAR: Probabilistic forecasting with autoregressive recurrent networks," *Int. J. Forecast.*, vol. 36, no. 3, pp. 1181–1191, 2020.

[10] D. K. Sharma, "The impact of e-commerce on operational cost efficiency in modern businesses," *Int. Peer-Rev. Open Access Multidiscip. J.*, vol. 9, no. 1, pp. 56–62, 2024.

[11] A. Shiri et al., "A comprehensive overview and comparative analysis on deep learning models," *J. Artif. Intell.*, vol. 6, no. 2, pp. 1–62, May 2024.

[12] X. Wang et al., "A novel feature attention mechanism for improving the accuracy and robustness of runoff forecasting," *J. Hydrol.*, vol. 617, p. 128916, 2023.

[13] S. Xu, J. Li, Y. Guo, and X. Zhang, "A parallel GRU recurrent network model and its application to multi-channel time-varying signal classification," *IEEE Access*, vol. 7, pp. 118739–118748, 2019.