National College of Ireland

Optimising Gated Recurrent Unit for Intrusion Detection in Internet of Things Networks: A Comparative Analysis with Other Deep Learning-Based Methods

MSc Research Project
Data Analytics

Jothybala Murugan

Student ID: x22245201

School of Computing

National College of Ireland

Supervisor: Dr. Abid Yaqoob

National College of Ireland Project Submission Sheet School of Computing

Student Name:	Jothybala Murugan
Student ID:	x22245201
Programme:	Data Analytics
Year:	2024
Module:	MSc Research Project
Supervisor:	Dr. Abid Yaqoob
Submission Due Date:	12/08/2024
Project Title:	Optimising Gated Recurrent Unit for Intrusion Detection in Internet of Things Networks: A Comparative Analysis with Other Deep Learning-Based Methods
Word Count:	7162
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	12th August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Optimising Gated Recurrent Unit for Intrusion Detection in Internet of Things Networks: A Comparative Analysis with Other Deep Learning-Based Methods

Jothybala Murugan

x22245201

Abstract

The research project's primary goal is to protect the Internet of Things devices from the cyber-attacks using the advanced neural networks. This paper was discussed about the increase in accuracy as well as effectiveness of the intrusion detection in Internet of Things (IoT) network by optimising the design and hyperparameter of Gated recurrent units (GRUs). This research work was fully dedicated to the growth of an attack classifier, which was the intrusion detection system. The rapidly increasing in the number of cyber-attacks have made intrusion detection prediction research essential. It is important to maintain the security as well as integrity of the Internet of Things (IoT). Even though for predicting the intrusions there are many ways to solve this problem but the most efficient way we used in this research to solve this is Gated Recurrent Units (GRUs) which is a gating mechanism in deep learning. In this deep learning technique, we have used a specifically used the Bidirectional GRU and convolutional gated recurrent unit (ConvGRU). The different network traffic data in the existing RT-IoT dataset, we evaluated the models based on different type of attack scenario. The gated recurrent unit's models were trained in this research and evaluated the model's accuracy using the IoT dataset. A deep learning model based on Gated Recurrent Units (GRUs) is explained and showed results can predict future alert probabilities from an attacking source. A comparison was evaluated using evaluation metrics and from the analysis the Convolution gated recurrent unit performed well than the other two models and detect the IoT attacks with a precision of 98%, F1 Score of 98.3% and an accuracy of 98% with minimum loss value of 0.04 respectively. Based on the evaluation, using the deep learning methods, we can conclude that the intrusion detection in the IoT can be solved successfully. This entire research handled with the ethical issues which include data confidentiality as well as reliability.

Keywords: Internet of Things, Gated Recurrent Units, intrusion detection, deep learning.

I. INTRODUCTION

Nowadays the internet of things has become widely used in our daily lives and the maintaining the IoT security is highly important for the IoT devices. The growth of Internet of Things (IoT) in the number of industries, including the hospitals, manufacturing, transportation and home automation, smart devices have increased so much more. With the help of the internet, these electronics devices are connected to the network. These devices were allowed for fast data transfer and these innovations that improve the daily ease and efficiency of the devices. Farooq., et al (2015) author analysed and reported that 25 billion were connected to devices in 2020. These networked devices improve daily tasks and provides innovative solutions. However, the major security issues and decisions that create high advantages and opportunities in the IoT technology provide. While developing IoT solutions, that the connected devices that needs to be controlled. The IoT devices are unprotected to different attack types of cyberattacks

because of their low power consumption, weak security protocols, and low understanding by users. By reviewing huge quantities of IoT data, we used deep learning and data analytics techniques to enhance customer service and network performance. In Internet of Things networks, pattern recognition, anomaly detection can be used to identify potentially dangerous behaviour. In this research, we have used the RT-IoT dataset which was a large dataset for IoT networks traffic analysis and shared by the authors Sharmila and Nagapadma (2023).

1.1. Background and Motivation

Deep learning with the help of neural networks algorithms can be used in wide range of applications including image recognition, speech recognition, computer recognition, etc. Improving the intrusion detection system can be achieved by this deep learning technique recurrent neural networks. This deep learning technique helps automatically learn from and extract information from this RT-IoT datasets. The Recurrent neural network which is a type of gated recurrent units (GRU). It helps in demonstrating the considerable potential in sequence in modelling tasks which is suitable for the network traffic analysis. In order to increase the intrusion detection in IoT network's accuracy and efficiency, gated recurrent units will be suitable for this intrusion detection.

1.2. Research Question and Objectives

Research Question of this study is outlined and described below.

- What are main issues that concerning the privacy and security in IoT networks.
- Does the optimized Gated Recurrent Units (GRU) architecture and hyperparameters tuning works in improving the accuracy and efficiency of intrusion detection in Internet of Things (IoT) networks.
- Does Gated recurrent unit model perform well than other deep learning models in terms of computational cost and performance.

The main goal of this research is to optimise the architecture and hyperparameters of Gated Recurrent Units (GRU), Bidirectional GRUs (BiGRU), and Convolutional GRUs (ConvGRU) in order to improve the efficacy and accuracy of intrusion detection in Internet of Things (IoT) networks. The goal of this thesis is to address the issues of the related to security and privacy in the IoT and the also aims not only for the better accuracy of this dataset, but also works effectively within the resource limitation characteristic of IoT environments by investigating different setups and optimisations of GRU-based models. Three goals were achieved in this research. Firstly, the optimized GRU will improve the GRU model in the context of IoT networks. Secondly, this research finds out how the hyperparameter will affect the GRU, BIGRU, ConvGRU model's performance. By systematically tuning the hyperparameter values including number of hidden units, batch size, etc and helps in identify the enhanced model's performance. Finally, a framework for implementing the optimised IDS models in actual IoT. This also testing models on the network traffic and evaluate performance metrics of the models. In the summary, intrusion detection systems accuracy and efficiency, this research compares

these models and deep learning techniques and solves the particular issues of the Internet of Things.

2. Related Work

2.1. Overview of machine learning and deep learning techniques in IDS.

Machine learning, data mining are widely employed to address the issues because to the rapid improvements in cyber-attacks and massive volumes of dangerous data in cyber infrastructures. Traffic on networks profiling, identifying anomalies, signature detection are all applications where machine learning can be used. In order to improve accuracy and efficiency, a lot of research has been done in the area of intrusion detection systems (IDS) Liao, et.al,.(2013) explained the Intrusion detection system. From this intrsion detection system for Internet of Things (IoT) networks, utilising advanced deep learning techniques. This section reviews some of the primary studies that have used Gated Recurrent Units (GRU), Bidirectional GRUs (BiGRU), and Convolutional GRUs (ConvGRU). Some past researchers mainly focused on the deep learning and neural networks. The field of intrusion detection system in IoT had been growth rapidly because of millions of data transfers. The various learning techniques from traditional machine learning to deep learning helps in predicting the Intrusion detection system. Particularly in network security there are effective new methodologies were explained by recent authors. In this section, the basis for this study is established, research gaps in the literature are noted, and past research findings are summarised. Abrar et al (2020) explained other machine learning techniques like random forest, Extra-tree classifier and decision tree using the NSL-KDD dataset. The results showed 99 percent for different features of attack classes and concluded with effective prediction rate.

Internet availability, integrity, and privacy are becoming increasingly of a concern nowadays due to the rapid increase in their use. Kaur, M., et al (2020) explained and the usage of single, hybrid and ensemble machine learning (ML) classifiers. They also evaluated with the seven different dataset and got the accuracy of about 90 percentage. A detailed review was also done by Yang, Z et al. (2022) and showed the accuracy of anomaly-based network intrusion detection techniques. They also suggested a model to evaluate IDS performance in various situations and indicated the significance of using a variety of datasets. In deep learning techniques, the Recurrent neural network (RNN) used for the sequential data and capturing temporal dependencies had led widespread adoption in the field of Intrusion Detection Systems (IDS). In comparison to other deep learning techniques like Long ShortTerm Memory (LSTM), the architecture of GRU, Recurrent neural
network version, is less complicated and uses fewer gates, which could result in shorter learni ng times and lower cost of computation. (SHAP)Shapley Addictive explanation by anywarg et al (2021) explained the anomalies detection in autoencoders that can be used in model predictions to provide insights and in making the models more clear and easily obtainable. Research by Mehmood et al. (2023) explained the GRU models wll be better than the other machine learning models and also they got the accuracy rate of 98.86 percentage. The significance of choosing the correct model for efficient feature learning in IDS is highlighted by this work.

2.2. Comparison of Deep learning techniques in Intrusion Detection.

In 2023, Luo, F., et al. explored on the gated recurrent unit (GRU) autoencoder network with emphasis on in-vehicle network intrusion detection systems. Their research showed the high accuracy rates of about 89 percentage and potential of GRU architectures to detecting the anomalous behaviours in vehicle networks. On the Kyoto 2019 dataset, Vinayakumar et al. (2019) demonstrated a deep neural network model that shown great accuracy. GRU models performed better than traditional techniques, with the evaluation metrics of precision, recall, f1-score and accuracy of above 88%, according to their comparison of various deep learning techniques. Also using the NSL-KDD 1999 dataset, Javaid et al (2016) explained a DL-based techniques which was self-taught learning (STL) to identify the network anomalies. In this technique, we can be able to gather attributes from numerous network sources. The malicious and normal traffic, these preprocessed data were processed through the auto-encoder and regression. Their analyses were better than other DL-based research. The DL-based Deep Neural Network (DNN) was about 15 percentage higher than the SVM approach. Sousa et al. (2023) developed an efficient IDS by using methods inspired by nature and intelligent intrusion detection system for 5g-enabled internet of vehicles. The approach they used greatly subject and how flexible models are in responding to different network environments. Their tests showed that the proposed IDS achieved F1 scores 1.00, 0.98 accuracy using decision trees and random forests in which no one had achieved using the Network Simulator 3 (NS-3). The authors Al-Imran et al. (2021) compared deep learning models with machine learning approaches using the Kyoto Honeypot Dataset. They used three phases, on the first phase, they used SVM, Decision Tree, and KNN. In the second phase, they used Random Forest, and XGBoost in which it showed significant improved than the first techniques in supervised learning. Finally, deep learning techniques, Feed Forward, LSTM, and Gated Recurrent Unit neural network were used in the last phase. They also found that these techniques consistently performed better than different algorithms. Their study made clear how crucial model selection is to getting the most effective possible IDS performance. The comparison between the ML techniques and deep learning techniques by Yin, C et al (2021) examined and compared recurrent neural networks models performance with other ML techniques in IDS. They also focused on binary classification and multiclass classification, number of neurons and different learning that impacts on the performance of the proposed models and concluded with RNN models performed well with high accuracy. Also they used the NSL-KDD dataset. Using Synthetic Minority Oversampling Technique and Random Forest algorithm algorithms, Alshamy et al. (2021) explained an IDS that achieved remarkable accuracy on the KDD dataset. The importance of feature selection in enhancing model performance was highlighted by their study. Also they used the SMOTE, Adaboost , Logistic Regression, and Support Vector Machine classifiers and evaluated the metrics based on accuracy, precision, recall, f1-score, and time for both binary and multi-class classification.

2.3. GRU, BIGRU and ConvGRU models Based IDS in IoT Networks

There are many choices for the training and evaluating the models using the datasets. The most used were NSL-KDD, KDD Cup 1999, benchmark dataset, CICIDS. In this research, we used

different dataset which was RT-IoT dataset. Sharmila et al (2023) collected the dataset and gave the new IoT dataset for the Intrusion detection systems. Using the NSL-KDD IoT dataset, Zhao et al (2023) explained the IoT based intrusion detection predictions using the Gated recurrent unit and Residual Network. The data features were trained using gated recurrent unit. The time series features of the sample data were collected and the data features were classified using ResNet, and the classification results are normalised using the function of softmax. They concluded that the IDS model achieved 96.12% accuracy and 97.85% detection rate, which was 1.86% and 2.59% higher than LSTM-ResNet. Kanika et al (2023) proposed a new hybrid deep learning models which was (LSTMVAE-BiGRU) the Long Short-Term Memory Variational AutoEncoders and Bidirectional Gated Recurrent Units explained that the BiGRU's model performance in detecting complex attack patterns in IoT networks. In this the author Cao et al (2022) explained and incorporated convolutional neural network and bidirectional gated recurrent unit for multiple classifications was proposed. In this paper, the proposed models were evaluated and the feature selection based on the random forest algorithm and Pearson correlation analysis in the UNSW_NB15, NSL-KDD, and CIC-IDS2017 datasets with an accuracy of 85.55%, 99.81%, and 99.70%, which was 1.25%, 0.59%, and 0.27% better than the CNN-GRU.

This research is mainly focused on optimising the GRU architecture and intrusion detection system hyperparameters and comparing its performance and cost against Bidirectional GRU (BiGRU) and Convolutional GRU (ConvGRU) models for real-time IoT networks. In order to handle the new security challenges, an enhanced intrusion detection systems (IDS) will be required in Internet of Things (IoT) environments due to the number and different network traffic types. In order to handle the new security challenges, an enhanced intrusion detection systems (IDS) will be required in Internet of Things (IoT) environments due to the number and different network traffic types. This deep learning GRU-based intrusion detection system compares with other GRU models as well based on its performance and computational cost provided an in-depth study of the advantages and disadvantages of these approaches for identifying cyber threats in IoT networks. In summary, the research on intrusion detection systems prediction showed that the internet grows quickly when deep learning and machine learning methods were combined. So, we have the used the most suitable models for this IDS which was GRU and the other combined GRU models.

3. Research Methodology

This research was focused on the building an optimised Gated recurrent unit and based on the performance compared with other GRU variants which were Bidirectional Gated recurrent unit and Convolutional gated recurrent unit. From the literature reviews we can be able to see that the GRU is one of the most suitable model architectures for the IDS. The IoT dataset used in this research and compared with three models for the better performance. In order to evaluate the performance of intrusion detection systems, it is important to collect numerical data in order to measure model accuracy, precision, recall, and F1-score of the models.

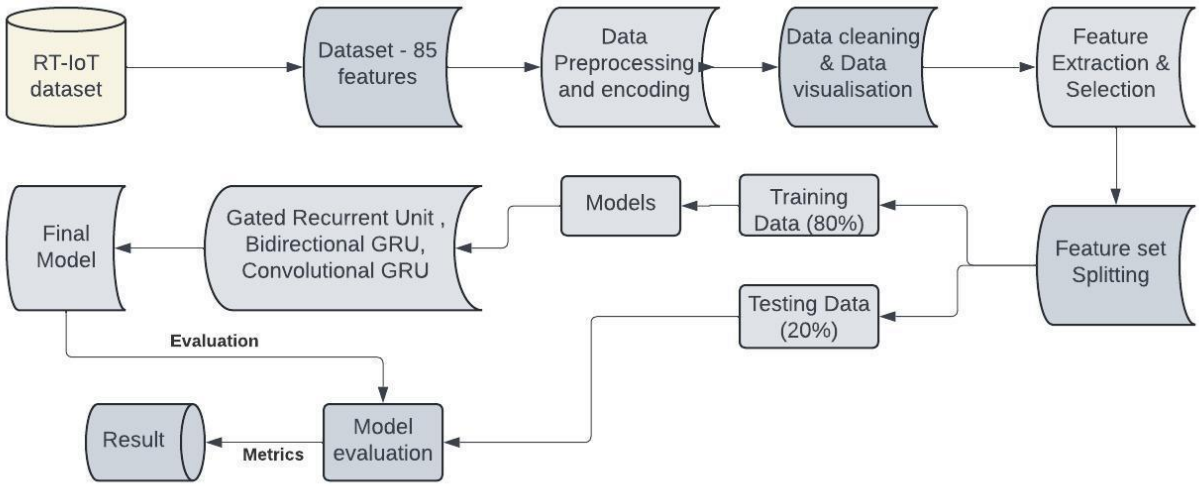

Figure 1: Flowchart of Methodology

3.1. Data Collection

The first step was collecting the relevant data. In this research, the RT-IoT2022 dataset was used. In this dataset there were huge number of network traffic data including both authorised and malicious attack types in the IoT devices. Firstly, the authors collected the dataset information from Internet of Things devices such as ThingSpeak-LED, Wipro-Bulb, and MQTT-Temp. The dataset also included details about the different types of malicious attacks that were conducted in the devices which including DDoS attacks, Brute-Force SSH attempts, and Nmap patterns. The authors initially had 12000 instances with 66 features. Now, they have collected 123117 instances with 85 features. This dataset has been obtained from the UCI repository. It is on open repository. The below following one is the URL for the dataset. https://archive.ics.uci.edu/dataset/942/rt-iot2022

3.2. Data Preparation

One of the most important steps in developing an efficient intrusion detection system (IDS) is data preparation. In this research, we used python programming with the necessary libraries. After collected the data, the next step was data preparation, and it was necessary to clean and preprocessing it for further analysis. The IoT dataset contains large numbers of traffic IoT data. The data is collected and converted into suitable format for the analysis. As this RT-IoT dataset contains 123117 rows and 85 features, the data was prepared and used for Pre-Processing and transformation process. As the features had more numbers of traffic data which includes the Timestamp, Source IP, Destination IP, Source Port, Source IP, Destination Port and destination IP, Protocol, Packet length, Payload length, Flags, Durations and Attack types. These are the

main features for this dataset. Timestamp is the time of the network packet captured. The source IP and the destination IP were the two important features in the dataset which the IP address of both sender and receiver. Similarly, for the source port and destination port was the port number of both sender and receiver. Protocols feature were the network protocols like TCP, UDP. Packet length and Payload length were the length of network packet and their payload length. The other feature flags count that determines the flags set in the network packet. Duration is the network session connection feature and finally the target which is Attack type that indicates whether the packet belongs to specified attack types.

3.3. Data Pre-Processing and Transformation

Firstly, we gathered and prepared the RT-IoT dataset from the UCI repository and then we process to next step which is data pre-processing. In the Data preprocessing we handled the null values and any missing values in the dataset, and then converting categorical into numerical variables for building models and normalised the numerical values. In this process, we handled them carefully. The missing values, categorical variables, numerical variables and continuous variables, outliers and data inconsistencies in the dataset.

```
Missing values:
Unnamed: 0              0
id.orig_p               0
id.resp_p               0
proto                   0
service                 0
                       ..
idle.std                0
fwd_init_window_size    0
bwd_init_window_size    0
fwd_last_window_size    0
Attack_type             0
Length: 85, dtype: int64
```

Figure 2: Null values

Firstly, we checked for the any missing values in the dataset. The dataset had no missing values which was shown in figure in 2 and so that we can be able to proceed with encoding the categorical variables, numerical variables and continuous variables. Secondly, the dataset has some categorical variables. 'proto', 'service', 'Attack_type' are the categorical variables in the dataset which has their unique values. After finding the categorical variables, we have decided to convert them into numerical variables. To convert them into numerical format we used label encoder technique here. As we used the label encoding for converting them in a numerical value for each unique value in the categorical feature, we have converted into numerical values successfully. Label encoding additionally helps in providing an implicit connection between two categories in an ordinal method. Then, we pre-processed the numerical variables and continuous variables and worked on it.

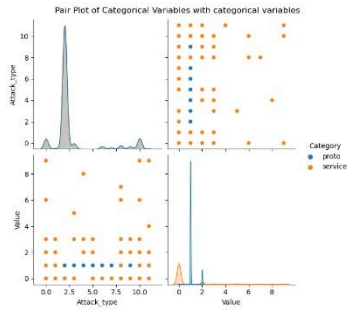

Figure 3: Pair Plot of Categorical Variables with attack types.

In figure 3, the pair plot showed the relationships between categorical variables and attack type numbers. The remaining features of 82 are numerical variables and continuous variables. Then we found that the transformed data had outliers. So, we decided to remove them using the Z-score threshold method and numpy absolute to remove them. After removing them, we proceed with the feature selection for preventing from the overfitting. We used four different use case for the features selections and each use cases helps in understanding the correlation matrix to find any multicollinearity in the features. Finally, the data preparation involved in transforming the data into suitable format for the three models. In this, we include splitting the dataset into training data, testing data and validation for tuning hyperparameters and the testing data is used for evaluating the model's performance.

4. Design Specification

For this purpose of Optimising Gated Recurrent Unit for Intrusion Detection in Internet of Things Networks three different GRU were chosen: Gated Recurrent Unit (GRU), Bidirectional GRU (BiGRU), Convolutional GRU (ConvGRU).

- **Gated Recurrent Unit (GRU)***:* In this model, used four layers in GRU with 128 units enabled with return sequence. In second layer with the 64 units. In the layer 3, 0.5 dropout rate and in the fourth dense layer with a unit equal to the number of classes and the activation function used is softmax.

- **Bidirectional GRU (BiGRU**): In this model, used the same four layers in BiGRU with 64 forward, 64 backward units enabled with the return sequence. Similarly, in second layer with 32 forward units and 32 backward units with the dropout rate of 0.5 in the layer 3 and in layer 4 used same softmax.

- **Convolutional GRU (ConvGRU):** In this model, we used the different one which is Conv1D with 64 filters with the kernel size of three. The activation function used is relu. On layer 2 used 128 units with the GRU. In dense layer 3 with the same dropout rate of 0.5. and in dense layer 4 used the softmax function.

Table 1: Proposed Model Architecture

Specification	**GRU Model**	**BiGRU Model**	**ConvGRU Model**
Input Shape	Timesteps + features	Timesteps + features	Timesteps + features
Layers	GRU + Dense+ Dropout	BiGRU+ Dense+ Dropout	Conv1D + GRU + Dense+ Dropout
Number of GRU Units	128 + 64	128 + 64 (forward & backward)	128 (GRU units after convolution)
Convolution Layers	None	None	Conv1D (filters=64+ kernel_size=3,+activation='relu')
Activation Function	Softmax (classification)	Softmax (classification)	Softmax (classification)
Loss Function	Categorical Crossentropy	Categorical Crossentropy	Categorical Crossentropy
Optimizer	Adam	Adam	Adam
Dropout Rate	0.5	0.5	0.5
Batch Size	32	32	32
Epochs	5-10 (early stopping)	5-10 (early stopping)	5-10 (early stopping)
Early Stopping	Monitor='val_loss'+ patience=3	Monitor='val_loss'+ patience=3 + restore_best_weights=True	Monitor='val_loss'+ patience=3
Validation Split	0.2	0.2	0.2
Metrics	Accuracy, Precision, Recall, F1-Score	Accuracy, Precision, Recall, F1-Score	Accuracy, Precision, Recall, F1-Score

In the above Table 1, the proposed models with the hyperparameter tuning are shown clearly. With the tensorflow libraries, the models were trained using various output parameters, such as Epochs and drop-out rates to determine the best model for the IDS.

5. Implementation

5.1. Setup

For the setting up process, used the computing environment with both training and building the models. Also, the proper hardware and software combinations is usually required for this. The deep learning models in this study were trained more quickly by using an advanced computing environment equipped with a Graphics Processing Unit (GPU). The usage of GPU in here is crucial for processing the large network IoT data that associated with training GRU and the other models for improving the model performance effectively. The software environment was chosen with care in addition to the hardware specifications. Python is our main programming language in this research because of its many machine learning and deep learning libraries we used them here. Also, for building and training the models some important libraries like TensorFlow and Keras were utilised. With the help of these libraries, neural network implementation part provides for the final evaluation process. In addition, NumPy and Pandas libraries, which provide strong capabilities for successfully handling the datasets, were used for handling data and analysis.

5.2. Data Handling

In the context of intrusion detection systems, an accurate data handling is important for the efficacy of any learning project and the model performance directly impact the data quality and integrity. The RT-IoT2022 dataset was gathered with 123117 rows and 85 features and analysed the traffic data to determine the features and architecture before the data implementation. Many features, such as timestamps, source and destination IP addresses, protocols, payload sizes, attack types were included in this dataset to describe the features of network traffic. After the collection of the RT-IoT dataset, the first step used in data handling was data preprocessing. In this step, we cleaned the data for removing any data inconsistencies. Also, there was no null values in the dataset. Then, we make sure the numerical features were in normalised. This step is crucial for models since it has major impact on the models' performance and rate of convergence. We have used the label encoding method to convert the categorical features into numerical format. There are 3 different categorical features 'proto', 'service', 'Attack_type'. We used Z-score normalization for the transforming the features to have a zero mean and a standard deviation of one. Following that in the feature selection process there are more number of features, so we decided to use different use cases in order to remove less important features. after this, for getting the better performance, the variables for each model, hyper-parameter tuning has been done in the models. Libraries for TensorFlow and Keras have been utilised appropriately. Additionally, we divided the dataset into testing and training sets. The models

were fitted using the training set, and their performance was evaluated using the testing set. we used the validation set to avoid overfitting and fine-tune the hyperparameters. The models were fine-tuned with each dropout rate of 0.5. Also, we have used the ADAM optimiser for optimising and also to coverage the models. We performed around 5-10 epochs for the better accuracy for the three models. In the final step, we evaluate the data and also determine the final parameters. The model which has the highest validation accuracy after training with those features is evaluated as better model in terms of performance.

6. Evaluation

In this part, output of the three models were evaluated. The models were trained to differentiate between adversarial and standard data using the preprocessed data. The evaluation metrics utilised for evaluating the model's performance include recall, accuracy, precision, and F1 score accuracyare all of the evaluation metrics that have been used in relation to the confusion matr ix. In the end, we used the comparison chart for the three models' evaluation metrics results and showed the better model in terms of performance as well. The True positive, True negative, false positive, False negarive were used for the evaluation metrics.

Accuracy is evaluated by comparing the actual result percentage to all scenarios, which is assessed in traffic data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision always provides the solution to the positive question. There is equation for the precision for evaluating them.

$$Precision = \frac{TP}{TP + FP} * 100$$

The recall rate referred to the percentage of attack type that the model can accurately predict. Furthermore, when it has high percentage, the model is less likely to be incorrectly classified.

$$Recall = \frac{TP}{TP + FN} * 100$$

When the model is carefully evaluated, the efficacy of the model is also determined by the F1 Score. As relation to harmonic measures of recall and precision, this F1-score is used.

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

The final process is to predict the target variable which is attack type using the models used in here. The results and evaluation of our research is to predict the intrusion detection system using deep learning models which we have used here. We have used the target variable in this project is attack type. The deep learning models we used in this research were optimised GRU, BiGRU and ConvGRU models. Based on the model's evaluation, we compared and evaluate the suitable

model for the IDS. In evaluating the metrics, we have used the four different use cases based on the selecting the important feature from the dataset.

6.1. Packet-Level Features – Case study 1

In this case study one, we used the packet level traffic data for the testing and training data. We used “bwd_data_pkts_tot", "fwd_subflow_pkts", "fwd_pkts_tot", "fwd_pkts_payload.min", "fwd_pkts_payload.avg", "fwd_pkts_payload.max", "Attack_type". Also, it is a multi-class classification problem, we used the softmax activation sparse categorical crossentropy loss. We prepared the data by assigning the target variable (Attack_type) to y and the features of the first case study one which are feature matrix to X. The split ratio of about 80 and 20 percentage for training and testing data. Then, used the standard scaler and reshape the input data to reduce the mean to 0, standard deviation of 1 for standardise the features and fit the input requirements of the models in the case study one. Using keras Sequential API, the models were developed. It consists of 128 and 64 units GRU layers. To prevent the overfitting, we provided 50% of the input units to 0 for the dropout layer. In the last dense layer, we used the softmax activation function. Then, compiled with the Adam optimizer and specified loss function. We used early stopping for stop training if the loss of validation does not improve for three consecutive epochs and trained on the training data for 10 epochs with 32 batch size. A validation split of 20% is used to monitor performance on unseen data during training and also measured the total training time.

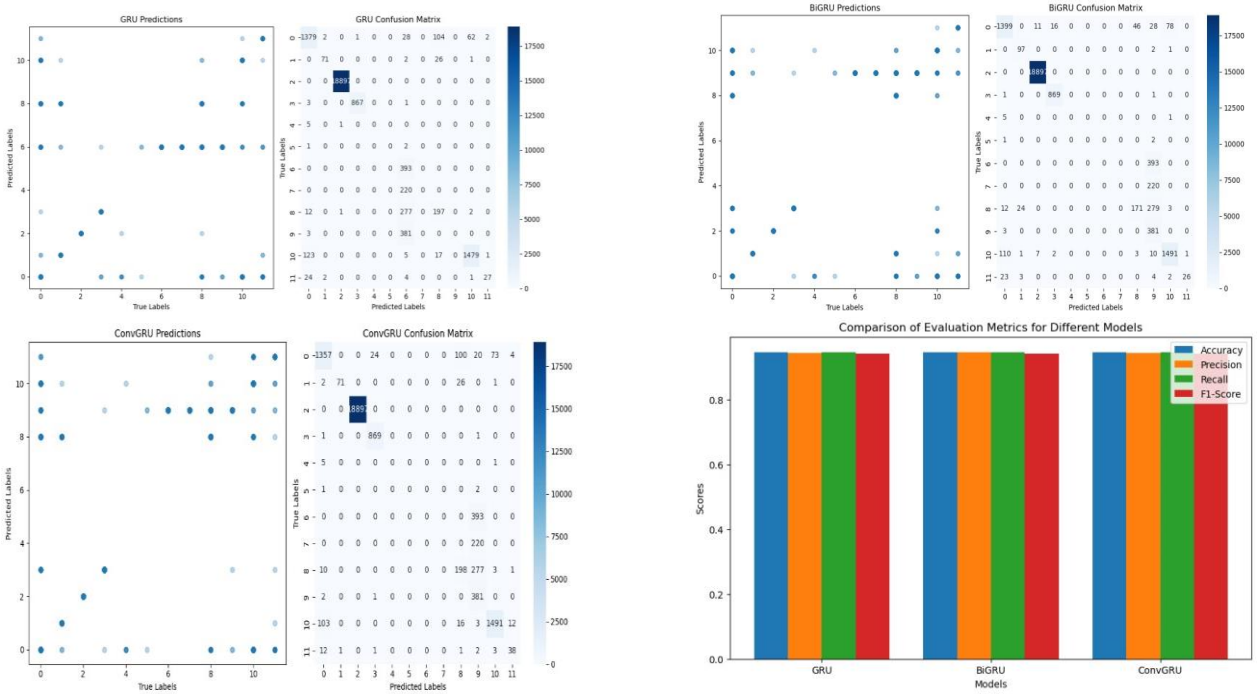

Figure 4: Confusion matrix and comparison chart of evaluation metrics

From the Figure 4, we can clearly see the confusion matrix and comparison of three models with their evaluation metrics including accuracy, precision, Recall, and F1score. From the first case features, the epoch of 10 and the loss 0.101 was settled for the three models. The evaluation metrics achieved 94% which is better test set accuracy.

Table 2: Classification Report of GRU model for case study 1

	Precision	Recall	F1-score	Support
1	0.89	0.9	0.91	1578
2	0.78	0.91	0.86	100
3	0.98	1	0.99	871
4	0.95	0.92	0.93	1625
5	0.96	0.45	0.61	58
Accuracy			0.94	24624
Macro Avg	0.55	0.55	0.52	24624
Weighted Avg	0.94	0.94	0.94	24624

In the above Table 2, output parameters for the Gated recurrent unit model were observed with respective classification report of accuracy 94%.

Table 3: Classification Report of BiGRU model for case study 1

	Precision	Recall	F1-score	Support
1	0.89	0.87	0.88	1578
2	0.95	0.71	0.81	100
3	0.57	0.41	0.47	489
4	0.96	0.91	0.93	1625
5	0.78	0.47	0.61	58
Accuracy			0.94	24624
Macro Avg	0.62	0.6	0.59	24624
Weighted Avg	0.94	0.94	0.94	24624

In the above Table 3, output parameters for the Bidirectional Gated recurrent unit model were observed with respective classification report with the accuracy of 94%.

Table 4: Classification Report of the ConvGRU model for case study 1

	Precision	Recall	F1-score	Support
1	0.91	0.86	0.88	1578
2	0.99	0.71	0.83	100
3	0.98	1	0.98	871
4	0.95	0.92	0.93	1625
5	0.69	0.66	0.67	58
Accuracy			0.94	24624
Macro Avg	0.53	0.52	0.54	24624
Weighted Avg	0.94	0.95	0.94	24624

From the above Table 4, output parameters for the Convolutional Gated recurrent unit model were observed with accuracy of 94%. We can see that the three models achieved 94% evaluation metrics.

6.2. Network Flow Metrics – Case study 2

In this case study two, we used the network flow traffic data for the testing and training data. We have used 'flow_duration', 'fwd_pkts_tot', 'bwd_pkts_tot', 'fwd_data_pkts_tot', 'bwd_data_pkts_tot', 'fwd_pkts_per_sec', 'bwd_pkts_per_sec', 'flow_pkts_per_sec', 'fwd_pkts_payload.min', 'Attack_type'. Also, this is multiclass classification problem. So used the same functions for the GRU, BIGRU and ConvGRU models and calculated the evaluation metrics, confusion matrix.

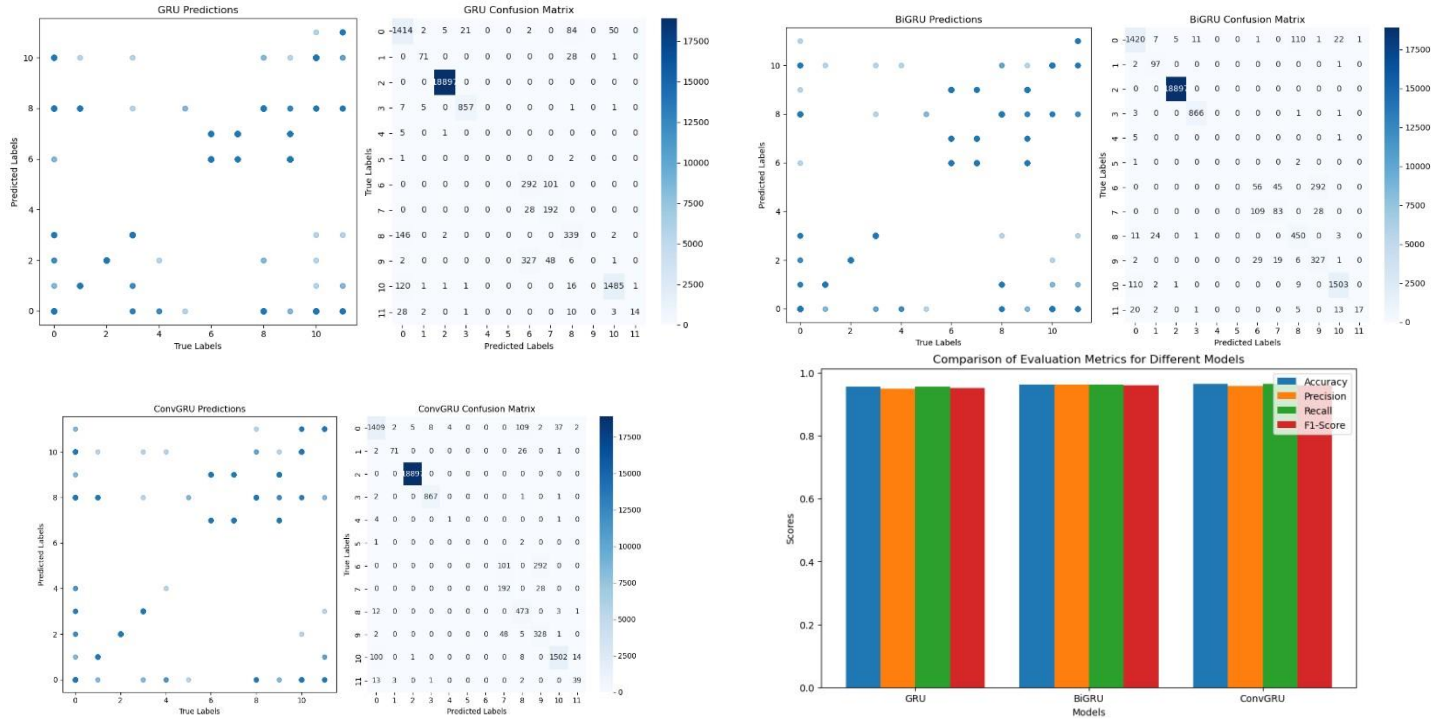

Figure 5: Confusion matrix and comparison chart of evaluation metrics of second case

From the Figure 5, we can be able to see the test accuracy of 97% which was better than previous case.

Table 5: Classification report of GRU model for case study 2

GRU	Precision	Recall	F1-score	Support
1	0.82	0.9	0.86	1578
2	0.88	0.71	0.78	100
3	0.97	0.98	0.98	871
4	0.96	0.91	0.95	1625
5	0.93	0.23	0.38	58
Accuracy			0.96	24624
Macro Avg	0.61	0.59	0.57	24624
Weighted Avg	0.95	0.96	0.94	24624

From the Table 5, the classification report of the gated recurrent unit model was determined, where we can see that the GRU model achieved with 96% by outperforming the first case.

Table 6: Classification report of BiGRU model for case study 2

BIGRU	Precision	Recall	F1-score	Support
1	0.9	0.9	0.89	1578
2	0.73	0.97	0.84	100
3	0.99	0.99	0.99	871
4	0.97	0.91	0.95	1625
5	0.94	0.3	0.45	58
Accuracy			0.96	24624
Macro Avg	0.64	0.61	0.6	24624
Weighted Avg	0.96	0.96	0.96	24624

From the above Table 6, the classification report of the bidirectional gated recurrent unit model achieved with similar 96% of the GRU model.

Table 7: Classification report of ConvGRU model for case study 2

ConvGRU	Precision	Recall	F1-score	Support
1	0.91	0.89	0.89	1578
2	0.93	0.71	0.81	100
3	0.99	0.99	0.99	871
4	0.97	0.92	0.95	1625
5	0.71	0.67	0.68	58
Accuracy			0.97	24624
Macro Avg	0.63	0.67	0.68	24624
Weighted Avg	0.97	0.96	0.96	24624

From the above Table 7, the classification report of the convolutional gated recurrent unit model outperformed with 97% than the other two models. From the second case, we can see that the convolutional gated recurrent unit performed well than the other two models

6.3. Network Traffic and Attack Analysis – Case study 3

In this case study, we used ‘fwd_pkts_payload.max', 'fwd_pkts_payload.tot', 'fwd_pkts_payload.avg', 'fwd_pkts_payload.std', 'bwd_pkts_payload.min', 'bwd_pkts_payload.max', 'bwd_pkts_payload.tot', 'bwd_pkts_payload.avg', 'bwd_pkts_payload.std', 'flow_iat.min', 'flow_iat.max’, ‘Attack_type' and used the same functions for the three models as before.

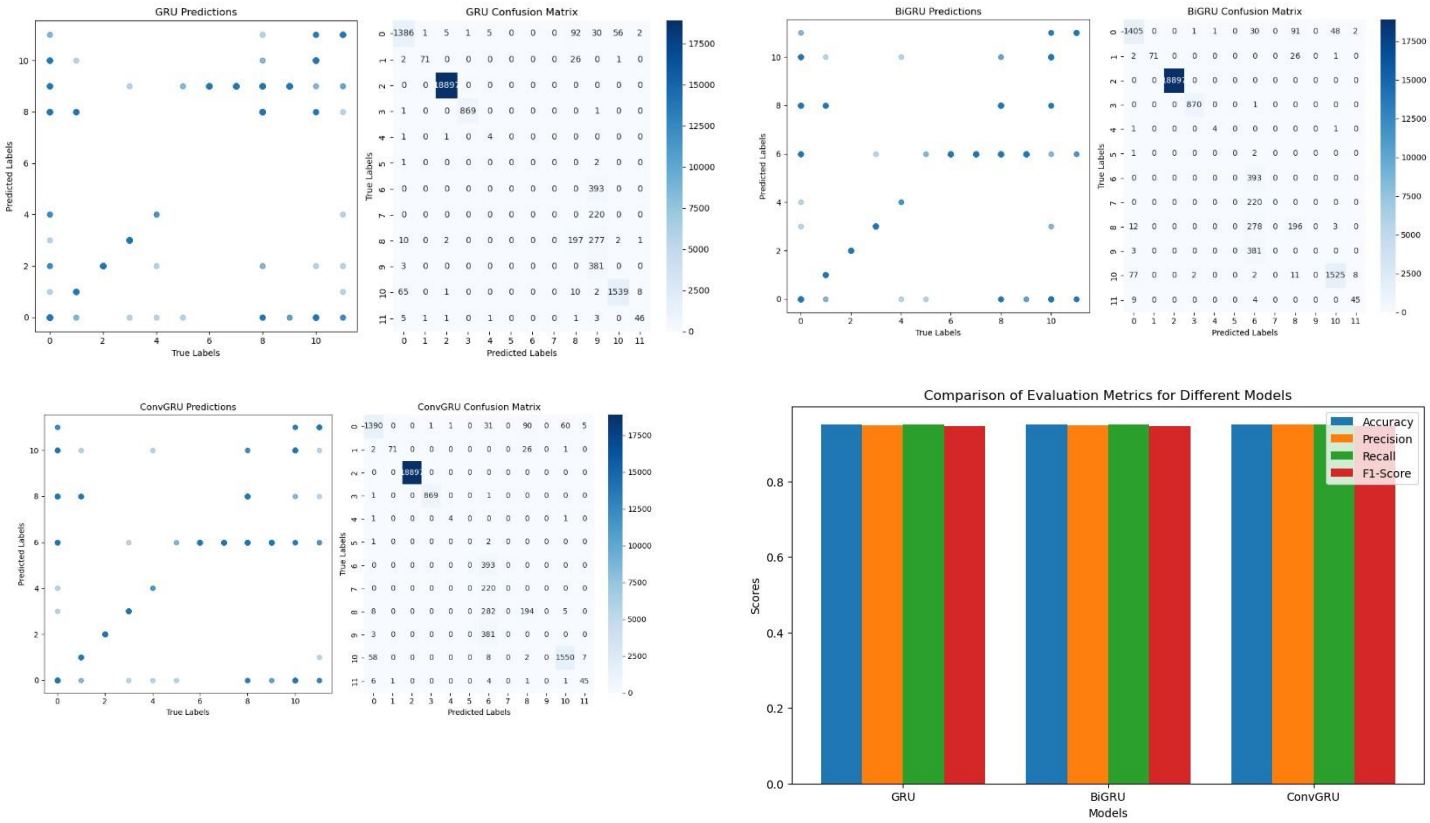

Figure 6: Confusion matrix and comparison chart of evaluation metrics of third case

From the Figure 6, we can be able to see the test accuracy of 95% and 0.05 loss was determined, which was lower than previous case.

Table 8: Classification report of GRU model for case study 3

GRU	Precision	Recall	F1-score	Support
1	0.94	0.88	0.91	1578
2	0.91	0.77	0.82	100
3	0.99	0.99	0.99	871
4	0.96	0.95	0.96	1625
5	0.81	0.79	0.8	58
Accuracy			0.95	24624
Macro Avg	0.58	0.62	0.59	24624
Weighted Avg	0.95	0.95	0.95	24624

From the Table 8, we determined the classification report of the Gated recurrent unit model with the f1-score, recall, precision and accuracy and support. From that, the GRU model performed with 95% test accuracy.

Table 9: Classification report of BiGRU model for case study 3

BIGRU	Precision	Recall	F1-score	Support
1	0.93	0.89	0.91	1578
2	1	0.71	0.83	100
3	0.99	0.99	0.99	871
4	0.97	0.95	0.96	1625
5	0.82	0.78	0.81	58
Accuracy			0.95	24624
Macro Avg	0.62	0.62	0.61	24624
Weighted Avg	0.95	0.95	0.95	24624

In the Table 9, the classification report of bidirectional gated recurrent unit model was determined, and we can be able to see that the Bidirectional gated recurrent unit model performed similar result with 95% test accuracy.

Table 10: Classification Report of ConvGRU using third case

ConvGRU	Precision	Recall	F1-score	Support
1	0.95	0.88	0.91	1578
2	0.99	0.71	0.83	100
3	0.99	0.99	0.99	871
4	0.96	0.95	0.95	1625
5	0.79	0.78	0.78	58
Accuracy			0.95	24624
Macro Avg	0.63	0.62	0.61	24624
Weighted Avg	0.95	0.95	0.95	24624

From the Table 10, the classification report of Convolutional gated recurrent unit model was determined, and we can be able to see that the Convolutional gated recurrent unit model achieved with similar two models with 95% test accuracy.

6.4. Network Traffic Flow and Payload Statistics – Case study 4.

In this case, we identified the most numerical features related to 'Attack_type' and visualize their interrelationships using the multicollinearity.

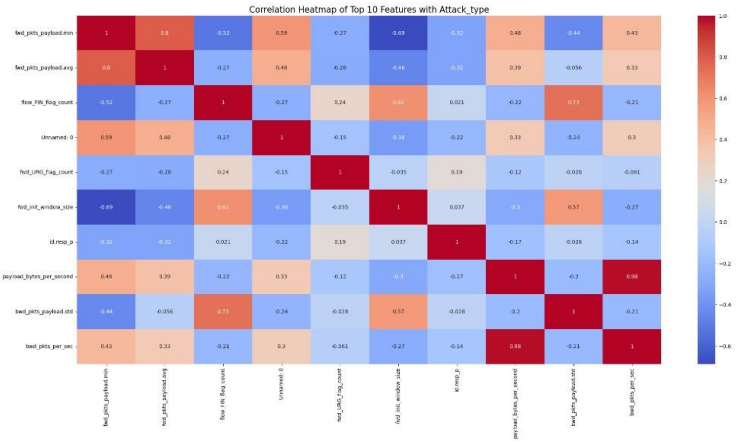

Figure 7: Correlation Heatmap of Top 10 Features with Attack_type.

From the Figure 7, We have used 10 top features which are identified features related to target variable and used the same function in this case also.

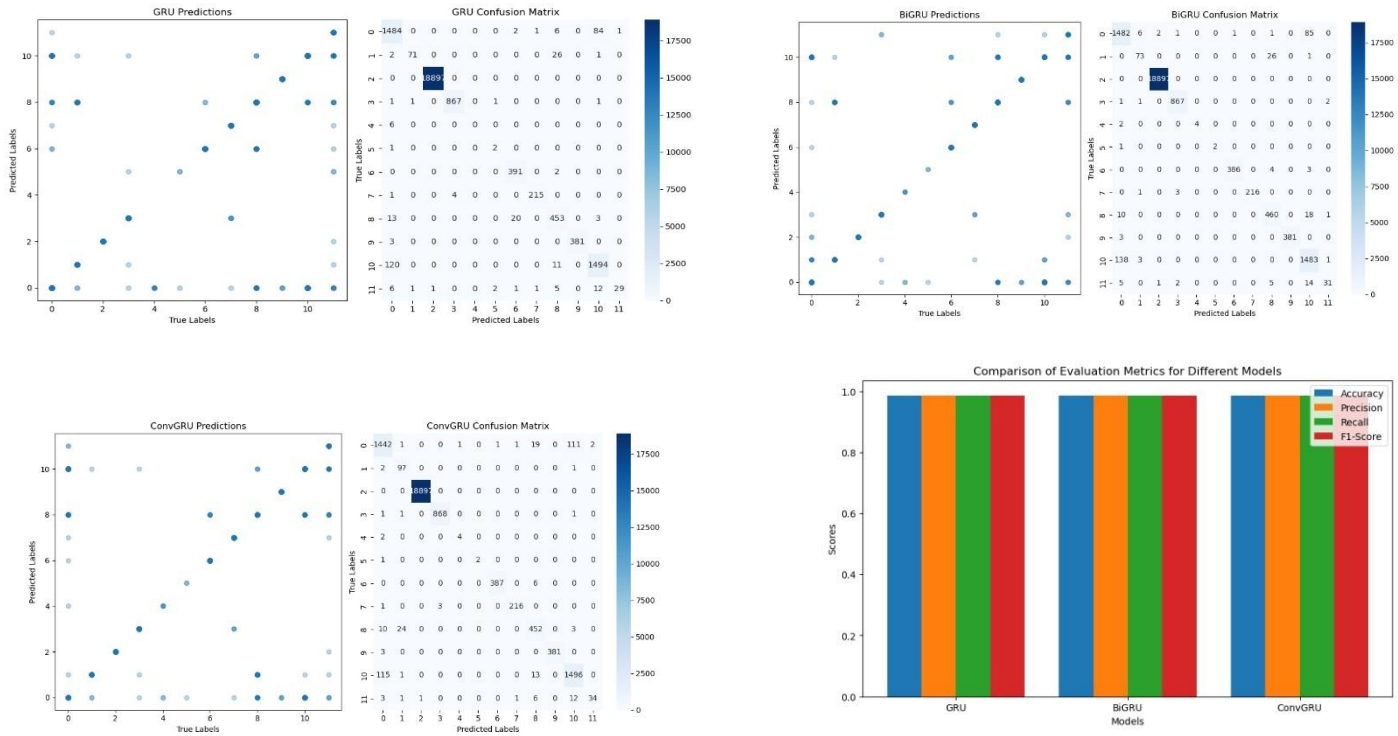

Figure 8: Confusion matrix and comparison chart of evaluation metrics of the fourth case.

From the above Figure 8, we can be able to see the test accuracy of almost 99% which was higher than other cases.

Table 11: Classification report of GRU model for case study 4

GRU	Precision	Recall	F1-score	Support
1	0.91	0.92	0.94	1578
2	0.97	0.71	0.82	100
3	0.99	0.99	0.99	871
4	0.94	0.92	0.93	1625
5	0.97	0.51	0.81	58
Accuracy			0.99	24624
Macro Avg	0.83	0.81	0.83	24624
Weighted Avg	0.98	0.99	0.99	24624

From the Table 11, we evaluated the classification report of the GRU model with the f1-score, recall, precision and accuracy and support and it achieved 99% of the metrics.

Table 12: Classification report of BiGRU model for case study 4

BIGRU	Precision	Recall	F1-score	Support
1	0.9	0.94	0.92	1578
2	0.87	0.73	0.79	100
3	0.99	1	0.99	871
4	0.92	0.91	0.92	1625
5	0.89	0.53	0.67	58
Accuracy			0.99	24624
Macro Avg	0.96	0.86	0.91	24624
Weighted Avg	0.99	0.99	0.98	24624

In the Table 12, we can also be able to look that bidirectional gated recurrent unit also achieved with the similar metrics of 99%.

Table 13: Classification report of ConvGRU model for case study 4

ConvGRU	Precision	Recall	F1-score	Support
1	0.91	0.91	0.91	1578
2	0.78	0.97	0.86	100
3	0.99	0.99	0.99	871
4	0.92	0.92	0.92	1625
5	0.94	0.61	0.78	58
Accuracy			0.99	24624
Macro Avg	0.94	0.88	0.91	24624
Weighted Avg	0.99	0.99	0.99	24624

From the Table 13, the classification report of Convolutional gated recurrent unit model was determined, and we can be able to see that the Convolutional gated recurrent unit model achieved with similar two models with almost 99% test accuracy and with the minimal loss of 0.04 than the three cases.

6.5. Discussion

This study aims to optimize the intrusion detection system and to reduce the intrusion from the malicious attacks. In this research project, we provided important findings and there are several directions of malware attack may happen in the cyber security professional.

Table 14: Comparison table of the GRU model on the four use cases.

GRU Model	Case 1	Case 2	Case 3	Case 4
Precision	0.95	0.96	0.95	0.98
Recall	0.95	0.96	0.94	0.99
F1-score	0.94	0.97	0.94	0.98
Accuracy	0.95	0.97	0.95	0.98

In the Table 14, we can clearly see the GRU model use cases was determined and evaluation metrics was achieved above 90% and we concluded the case 4 achieved the 99% accuracy, precision, Recall and F1-score.

Table 15: Comparison table of the BiGRU model on the four use cases.

BIGRU Model	Case 1	Case 2	Case 3	Case 4
Precision	0.94	0.97	0.96	0.985
Recall	0.95	0.96	0.94	0.98
F1-score	0.95	0.96	0.93	0.99
Accuracy	0.95	0.97	0.95	0.98

From the Table 15, the bidirectional gated recurrent unit model also evaluated with the similar performance of gated recurrent unit model. In this, the case 4 outperformance is better than all the cases.

Table 16: Comparison table of the ConvGRU model on the four use cases.

ConvGRU	Case 1	Case 2	Case 3	Case 4
Precision	0.95	0.96	0.95	0.99
Recall	0.94	0.96	0.94	0.99
F1-score	0.94	0.97	0.94	0.99
Accuracy	0.91	0.97	0.95	0.99

From the Table 16, the comparison table of Convolutional gated recurrent unit model was determined, and from that we can be able to see that the Convolutional gated recurrent unit model achieved with with almost 99% test accuracy on the fourth case.

We used the various learning rates, epochs, batch sizes, adam optimisers for this achieving the high-test accuracy results. We observed that the features selection process in this research provided different output. In the conclusion, the optimised gated recurrent unit model outperformed well and high accuracy in the predicting the malicious attack types on the internet of things. Also, the comparative analysis of the other two models which used in this research where bidirectional gated recurrent unit and Convolutional gated recurrent unit also achived like the gated recurrent unit model.

7. Conclusion and Future Work

In this study, an optimised Gated Recurrent Unit (GRU) model for intrusion detection systems (IDS) was proposed and evaluated and its performance was compared with the bidirectional GRU (BiGRU) and convolutional GRU (ConvGRU) architectures. The research highlighted that optimising the model performance requires of data preparation, which includes feature selection, data cleaning, and normalisation. After evaluating the metrics like precision, recall, and F1-score, the models outperformed well, achieved a high accuracy rate without overfitting.

Adding more features like network topology and device characteristics to the dataset is one such approach. This may enhance the model efficiency in detecting the cyberattacks. The models may become more efficient in finding the attack type if they consist of larger amounts of data, such as network traffic data and assessments of user behaviour. Moreover, enhancing the traditional machine learning classifiers and other deep learning architecture models for the real-time data streams is engaging because of the traffic data. Also, the focus on optimizing machine learning and deep learning algorithms reduces the latency and the speed of anomaly detection which can be improved. Optimising the model applicability in dynamic network instances may involve using the learning algorithms that may improve the models constantly as new data occurs. The interpretation of deep learning models in the Ids is important for nowadays because investigating the traffic data. Understanding the learning methods and their models helpful in cyber security professionals trust. Other techniques like (LRP)layer wise relevance propagation and Explainable Artificial Intelligence and LIME (Local Interpretable Model-agnostic Explanations) which can also be used for the clear understanding of the model behaviour and helps in cyber security professionals to validate the decision made by the Intrusion detection system. Finally, the future of the intrusion detection system has grown with new ways for innovation and improvements in cyber security and analytics. The advanced machine learning and deep learning and the combination of the new techniques, real time data processing, improved features engineering techniques, integration of threat intelligence will enhance the efficacy of the intrusion detection system models. These techniques not only improve detection but also provide important components in the cyber security for ensuring the network traffic data more safe and secure.

References

Abrar, I., Ayub, Z., Masoodi, F. and Bamhdi, A.M., 2020, September. A machine learning approach for intrusion detection system on NSL-KDD dataset. In *2020 international conference on smart electronics and communication (ICOSEC)* (pp. 919924). IEEE.

Aggarwal, K. and Kaddoum, G., 2023, December. LSTM-based hybrid intrusion detection system for Internet of Vehicles. In *GLOBECOM 2023-2023 IEEE Global Communications Conference* (pp. 3831-3836). IEEE.

Al-Imran, M. and Ripon, S.H., 2021. Network intrusion detection: an analytical assessment using deep learning and state-of-the-art machine learning models. *International Journal of Computational Intelligence Systems*, *14*(1), p.200.

Al-kahtani, M.S., Mehmood, Z., Sadad, T., Zada, I., Ali, G. and ElAffendi, M., 2023. Intrusion detection in the Internet of Things using fusion of GRU-LSTM deep learning model. *Intelligent Automation & Soft Computing*, *37*(2).

Alshamy, R., Ghurab, M., Othman, S. and Alshami, F., 2021. Intrusion detection model for imbalanced dataset using SMOTE and random forest algorithm. In *Advances in Cyber Security: Third International Conference, ACeS 2021, Penang, Malaysia, August 24–25, 2021, Revised Selected Papers 3* (pp. 361-378). Springer Singapore.

Antwarg, L., Miller, R.M., Shapira, B. And Rokach, L., 2021. Explaining anomalies detected by autoencoders using Shapley Additive Explanations. Expert systems with applications, 186, p.115736.

Cao, B., Li, C., Song, Y. and Fan, X., 2022. Network intrusion detection technology based on convolutional neural network and BiGRU. Computational Intelligence and Neuroscience, 2022(1), p.1942847.

Farooq, M. U., Waseem, M., Khairi, A., & Mazhar, S. (2015). A critical analysis on the security concerns of the internet of things (IoT). International Journal of Computer Applications, 111(7).

Hariharan, S., Rejimol Robinson, R.R., Prasad, R.R., Thomas, C. And Balakrishnan, N., 2023. XAI for intrusion detection system: comparing explanations based on global and local scope. Journal of Computer Virology and Hacking Techniques, 19(2), pp.217-239.

Javaid, A., Niyaz, Q., Sun, W. and Alam, M., 2016, May. A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)* (pp. 21-26).

Liao, H.J., Lin, C.H.R., Lin, Y.C. and Tung, K.Y., 2013. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, *36*(1), pp.1624.

Luo, F., Wang, J., Zhang, X., Jiang, Y., Li, Z. and Luo, C., 2023. In-vehicle network intrusion detection systems: a systematic survey of deep learning-based approaches. *PeerJ Computer Science*, *9*, p.e1648.

Montavon, G., Binder, A., Lapuschkin, S., Samek, W. And Müller, K.R., 2019. Layerwise relevance propagation: an overview. Explainable AI: interpreting, explaining and visualizing deep learning, pp.193-209.

Musa, U.S., Chhabra, M., Ali, A. and Kaur, M., 2020, September. Intrusion detection system using machine learning techniques: A review. In *2020 international conference on smart electronics and communication (ICOSEC)* (pp. 149-155). IEEE.

Sharmila, B. S., & Nagapadma, R. (2023). RT-IoT2022. UCI Machine Learning . https://doi.org/10.24432/C5P338.

Sousa, B., Magaia, N. and Silva, S., 2023. An intelligent intrusion detection system for 5g-enabled internet of vehicles. Electronics, 12(8), p.1757.

Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A. and Venkatraman, S., 2019. Deep learning approach for intelligent intrusion detection system. *Ieee Access*, *7*, pp.41525-41550.

Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y. and Han, H., 2022. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security*, *116*, p.102675.

Yin, C., Zhu, Y., Fei, J. and He, X., 2017. A deep learning approach for intrusion detection using recurrent neural networks. *Ieee Access*, *5*, pp.21954-21961

Zhao, G., Ren, C., Wang, J., Huang, Y. and Chen, H., 2023. IoT intrusion detection model based on gated recurrent unit and residual network. *Peer-to-Peer Networking and Applications*, *16*(4), pp.1887-1899.