JCBD
JOURNAL OF COMPUTERS AND DIGITAL BUSINESS
Volume 4, Number 2, May 2025, pp. 76-89
Homepage: https://jurnal.delitekno.co.id/index.php/jcbd

Network Log Implementation for GRU Based Bandwidth Classification

Azriel Christian Nurcahyo[1*], Huong Yong Ting[2], Abdulwahab Funsho Atanda[3]

[1] *Doctor of Philosophy (PhD) in Computing, University of Technology Sarawak, 96000, Malaysia*
[2,3] *Design and Technology Centre, School of Computing and Creative Media, University of Technology Sarawak, 96000, Malaysia*

* pic24030001@student.uts.edu.my

https://doi.org/10.56427/jcbd.v4i2.763

ARTICLE INFO

Article History
Received: March 3, 2025
Revised: May 9, 2025
Accepted: May 25, 2025

Keywords
Bandwidth
Fake Bandwidth
GRU
Genuine Bandwidth

ABSTRACT

Network bandwidth management using log data is a challenging task, especially in anomaly detection, e.g., fraudulent bandwidth that violates the Service Level Agreement (SLA). The present study suggests a deep learning automatic classification method for network logs, which leverages the Gated Recurrent Unit (GRU) and is used in time-series tensor configurations given as [N, 5, 15]. Data was gathered in real time during 29 days with the aid of a MikroTik RB1100AHx router, and it created more than 867,000 rows of data with three logs per second. The logs were classified into three classes: Genuine, Fake, and No Heavy Activity. Pre-processing involved windowing sequences, normalisation, and SMOTE balancing, whereas the GRU model comprised update and reset gates, followed by a Dense layer and a Softmax 3-class output. The model was trained with categorical cross-entropy loss and optimized with the Adam optimizer, validated with a 5-fold cross-validation strategy. The results achieved a 86.8% mean accuracy and an F1 score of 0.90 in the classification of Genuine Bandwidth, indicating that the GRU can successfully detect temporal patterns in network logs. This system is locally deployable through the G-Radio interface, demonstrating its feasibility, scalability, and substantial contribution to automatic bandwidth classification without packet inspection.

1. Introduction

Based on reference [1], traffic is one of the most important aspects in managing a company's information technology infrastructure in today's digital world. IoT-enabled devices, Internet users, and the need for cloud services for real-time connections like video streaming make bandwidth management a crucial problem in public and private networks [2]. Identifying and reliably classifying the usage of the network distinguishing between clear, non-standby, and idle periods is one of the prime issues an ISP faces in automating bandwidth management [3]. In ISP services for high speed Internet, the utilisation of network logging documents as the principal source of evidence for the performance of the network can be tremendously useful as it provides actual, continuous, and time-series data [4]. This study addresses these problems by building a bandwidth classification based on network logs using the deep learning method of Gated Recurrent Unit (GRU). To date, missing bandwidth has not been identified in great detail. Moreover, the absence of some bandwidth in this study is assumed to be fake bandwidth, which means the state the network is in where it does not respond through the bandwidth it is expected to deliver as stipulated in the service level agreement (SLA).

Beyond traditional approaches such as inspecting the TCP/IP layer or static statistical methods for identifying computer networks from QoS bandwidth values [5],[6], this study uses MikroTik RB1100AHx log data that records network status every second for 29 days. Previous similar studies have used machine learning for network traffic classification with decision tree algorithms [7],[8] and even Random Forest for bandwidth prediction [9],[10]. Viewed from this perspective, this approach faces the problem of capturing the temporal relationships (time sequences) of network logs, which are critically needed in real time network monitoring

systems. In this research, network logging is conducted over 29 days, 24 hours non stop from real time data monitored via Telegram.

A classification-based approach for detecting attacks in networks using anomaly detection has been implemented with LSTM (Long Short-Term Memory) in several studies [11],[12],[13]. However, this research does not classify traffic bandwidth. Other studies indicate that LSTM has weaknesses in terms of structure due to its high computational complexity and overfitting on large datasets [14],[15]. In this research, the GRU method, with a simpler architecture compared to LSTM, and comparable performance, is identified as a solution [16], making it ideal for local deployment with limited resources on servers [17]. Several studies suggest that GRU outperforms in classifying temporal sequences in sensor data and traffic from network computers [18],[19], but it has not yet been applied to the bandwidth logs generated automatically from enterprise-class MikroTik routers, whether on dedicated or dynamic bandwidth. This research also combines various technologies that are rarely found together in a single study, such as automated scripting in MikroTik, log transmission using SMTP and Telegram Bot, as well as pre-processing using windowing and SMOTE for Python based classification up to Python-based classification visualisation results. The study conducted by Haoming Wang reminds us how crucial pre-step and balancing data are in multi-class time-series classification [20]. On the other hand, reading log data for classification still heavily relies on parsing, whether manual or in specific formats provided from open source, or from the operating system and router's built-in formats [21],[22]. The approach attempted in this research uses normalized real-time input tensors of size [N,5,15] coming directly from the GRU architecture.In another literature, in studies such as [23], [24] and [25], fake traffic classification usually employs pattern attack detection but very few distinguish fake bandwidth in real time.

In institutional environments that subscribe to ISP services such as typical network subscription conditions in Indonesia a standard arrangement for a 200 Mbps connection often includes a Service Level Agreement (SLA) guaranteeing 10% to 20% of the total bandwidth. This means the service provider guarantees that the bandwidth upload and download will not fall below 20 Mbps to 40 Mbps. Any bandwidth performance falling beneath this threshold is considered as fake bandwidth or may be classified as lost bandwidth, rendering it undetectable in standard monitoring systems. In this study, to minimise classification errors during detection, a more stringent threshold is adopted, namely 10% of the total bandwidth. This implies that any bandwidth below 20 Mbps is used as the baseline for distinguishing between Fake and Genuine categories, whether the total bandwidth is 200 Mbps or even 300 Mbps. For operational purposes, the study defines the following classifications: fake bandwidth is specifically designated as bandwidth below 20 Mbps; genuine bandwidth is classified within the range of [21–100 Mbps]; and No Heavy activity is defined as below 5 Mbps. A constant bandwidth load of 200 Mbps was maintained continuously over 29 days, substantiated by uninterrupted download and upload activities for 24 hours each day. The network was accessed daily by an average of 20 to 40 users, with daily download activity exceeding 200 GB and upload exceeding 50 GB. It is anticipated that with these more detailed parameters and a substantially larger dataset, the analysis conducted in this study will offer a more objective basis than those found in previous literature.

Cheng Feng et al. and Shuo Zhang et al. consistently show in this research that the time-series of log data correlates strongly with the ability to identify abnormalities in cyber-physical systems, improving the illustration of captured log data [26],[27]. Meanwhile, Krzysztof Zarzycki et al. state that for dynamic phenomena that are not amenable to classic statistical methods, at least they can be explained by combining several sequence modelling techniques such as GRU and LSTM [28]. Bin Xia et al. emphasise the significance of time granularity and categorisation of log files according to access frequency and data transfer rate in reducing false positives in anomaly detection [29].

A study by Xiaocheng Huang et al. demonstrates that deep learning models such as GRU can outperform classical models not only in terms of accuracy but also in time efficiency when deployed on edge systems or servers with limited specifications [30]. In another study, Ashish Raghuwanshi et al. propose an embedding-based log event representation method to capture the contextual dependency among log lines, thereby improving the classification of network traffic at the enterprise level [31]. Meanwhile, the issue of model resilience to log noise is also raised by Jiangming Li et al., who show that raw logs from network devices contain many potential parsing errors, necessitating strict preprocessing that includes not only normalisation but also pattern match-based filtering such as on unmatched lines that are out of model in this research. Additionally, the development of a real-time classification system in this study is inspired by the streaming analytics system created by Basharat Hussain et al., which constructs an online inference system using IoT data with TensorFlow, where the GRU architecture has been proven to be optimisable for low latency [32]. In terms of class distribution dimensions, the SMOTE approach for balancing the minority class is also supported by Shujuan Wang et al. [33], where Wang showed that nearest neighbour-based interpolation in time-series domains yielded more stable classification results than ordinary oversampling. The GRU model is also more relevant in the context of logs with bandwidth fluctuations due to their inherent ability to more effectively address the vanishing gradient

problem than plain RNNs, as discussed by Amin Faraji et al. [34]. Furthermore, Dalia Abdulkareem Shafiq et al. demonstrated that identifying the temporal dimension for server log data could very accurately detect shifts in resource usage patterns and has optimisation implications for load balancing and SLA enforcement [35]. This underscores the contribution of this research in evaluating ISP SLA compliance based on actual bandwidth intensity as [36] suggests.

This study stands out in utilizing more than 867,000 records of actual network log data from the University of Technology Sarawak. This was translated from graphical to text format using a special configuration on the RB1100AHx router, not yet documented in the existing literature. This study stands out in being experimental in nature with actual settings. The primary goal of this study is to carry out training, tuning, and testing independently without the use of expensive cloud services. Cross validation accuracy consistently above 86%, corroborated by F1 of the Genuine class being 0.90, thereby validating the efficacy of the GRU model when applied to manually gathered network logs. This was so in spite of the prolonged processing time, which took seven consecutive days of uninterrupted execution. This work strives to bridge a significant gap that exists between packet based network classification research and bandwidth consumption centered research. The primary contribution is the process of ongoing log collection, organized into pre prepared time series tensors for GRU processing, and the application of real time data that can be processed within the same day or built up from continuously captured per second records. This is backed by the creation of a graphical user interface in Python and Jupyter Notebook, and the operational confirmation of more precise fake bandwidth detection using this methodology. These results pave new avenues for upcoming development in the field of bandwidth classification research.

2. Research Methodology

The GRU model in this study was chosen for implementation as a continuation of our previously published research on bandwidth classification using LSTM. However, further development was needed through a more concise and accessible method that does not require an excessive number of training epochs, yet is capable of handling larger volumes of data; thus, we decided to explore the GRU method [37]. Through this model, the researcher will attempt to achieve the goal of designing, implementing, and evaluating a bandwidth classification system based on network logs using deep learning based on Gated Recurrent Unit (GRU) techniques in this research method. This method is divided into two parts: the first part is network computing, where analysis is carried out by collecting network logs in real-time for 29 days, processing data, classifying traffic conditions, and implementing GRU; and the second part is deep learning for classification. The data in this study was acquired using a MikroTik RB1100AHx device connected to the public network of UTS Campus via IP 60.53.x.x/26 with direct access to unlimited dedicated bandwidth. The network logs were obtained at one-second intervals with three data entries stored in a Telegram bot for monitoring, and were automatically sent each day at noon for access from the mail server.

During the 29 day period, we successfully accumulated over 867,000 lines of log data that depict the actual traffic conditions on the Lab 4 network at University of Technology Sarawak. This network has continuously been installed with a bandwidth measurement tool with a maximum limit of 20 Mbps, where points below 20 Mbps are referred to as fake bandwidth or illegal access, from a total speed of 200 Mbps (static). Points taken from the bottom 10% are also considered lower stripe chunks of bandwidth or fake bandwidth. Furthermore, anything below 1 Mbps is classified as unmeasured, whether as fake or genuine, but generally falls within the criteria of no heavy activity. The data collected during this period includes connection time, type of bandwidth, and an estimate of whether the bandwidth is above or below estimate for each criterion. The internal bandwidth log configuration serves to send real-time proof of bandwidth logs. The bot can be monitored using the Telegram bot named @routeruts1100ahx_bot, subscribing robotically every second, as demonstrated in the real data collection method, as shown in Figure 1.

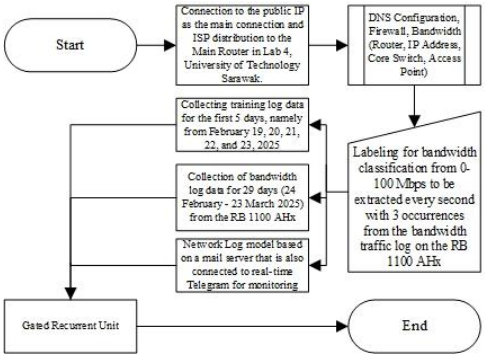


Figure 1. Data Collection Model from RB 1100 AHx

Furthermore, the data collection is recorded and sent through a Telegram broker that functions to receive telegrams from the mail server phdscm01@uts.edu.my as indicated in Figure 2, along with the journal log data that captures real activity user monitoring. After the data is collected, it undergoes cleaning which is commonly referred to as pre-processing, which involves modifying the input to make it suitable and usable by the GRU model. This also includes changing formats and normalising numerical data, categorical encoding, and time series transformation. Time series processing is carried out using windowing (cutting) with a duration of 5 seconds (5 data points per window) and sliding 1 second. Each window has 15 features, resulting in an input tensor of size [N, 5, 15]. Classification is indicated by the average bandwidth per window. The labelling criteria are set as follows:

1. Fake Bandwidth**,** if all bandwidth values in interval [t,t+2] are less than 20 Mbps

$$If\ \forall bi \in \{b(t), b(t+1), b(t+2)\},\ bi < 20 = Class = Fake\ Bandwidth$$

2. Genuine Bandwidth, if the average bandwidth over the window [t,t+4][t, t+4][t,t+4] lies within 21 to 100 Mbps

$$If\ \mu t: t+4 \in [21{,}100] = Class = Genuine\ Bandwidth$$

3. No Heavy Activity**,** if the average bandwidth over the same window is less than 5 Mbps

$$If\ \mu t: t+4 < 5 = Class = No\ Heavy\ Activity$$

Mathematically, the average bandwidth is calculated as: $\mu t: t+4 = \frac{1}{5}\sum_{i=t}^{t+4} b_i$

Furthermore, the bandwidth class distribution is unstable.According to the flowchart in Figure 2, most of the data is classified into the Genuine Bandwidth class. In order to balance the distribution, the SMOTE technique (Synthetic Minority Oversampling Technique) is applied. SMOTE generates new samples by interpolating between a minority sample and its neighbours using the formula

$$xnew = xi + \lambda(xneighbour - xi), \lambda \in [0{,}1]$$

Then the classes (Fake, Genuine, No Heavy) are encoded to a numeric vector form using One Hot Encoding, where Fake Bandwidth is valued at [1,0,0], Genuine Bandwidth at [0,1,0], and No Heavy at [0,0,1].

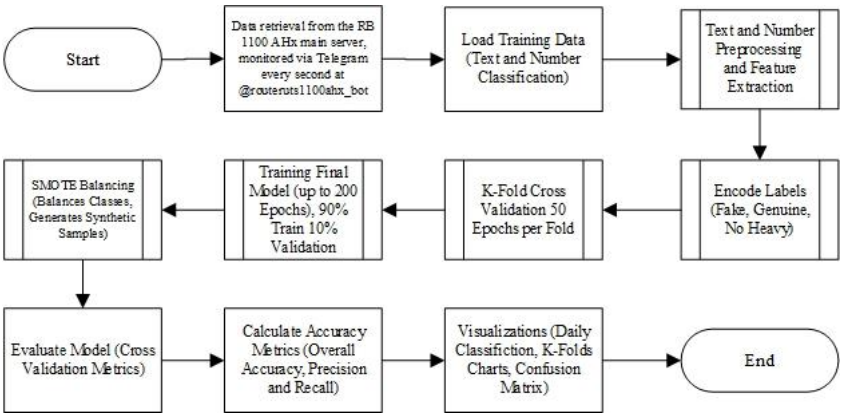


Figure 2. Bandwidth Classification Model

After that, the GRU process was carried out as illustrated in Figure 3, where the GRU (Gated Recurrent Unit) is a specialization that can accept input in the form of time-series by storing memory from the previous hidden states. In the GRU, there are two main gates: the update gate (z_t) and the reset gate (r_t) which are each expressed as

$$z_t = \sigma(w_z x_t + u_z h_t - 1)\ and\ r_t = \sigma(w_r x_t + U_r h_t - 1)$$

The final output of the GRU is connected to a Dense Layer, Dropout (0.3), and to a Softmax Layer with 3 neurons,

$$\hat{y}_i = \frac{e^z i}{\sum_{j=1}^{3} e^{z_j}}$$

Then the model was trained with a categorical cross entropy loss function,

$$L = -\sum_{i=1}^{3} y_i \log(\hat{y}_i)$$

Next, optimization was performed using the Adam Optimizer (α = 0.001), maximum Epoch = 200, Validation: 10% of the data. Cross-validation with 5 folds was used to test the generality of the model as shown on figure 3.

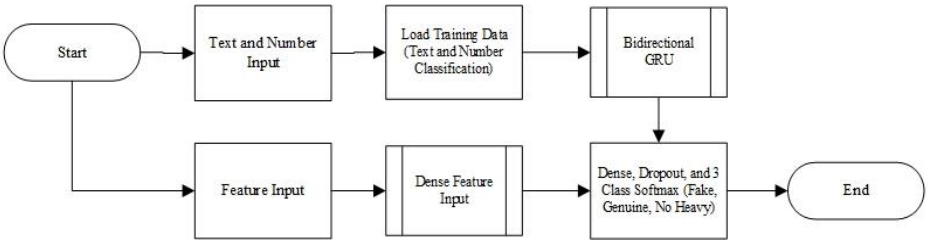
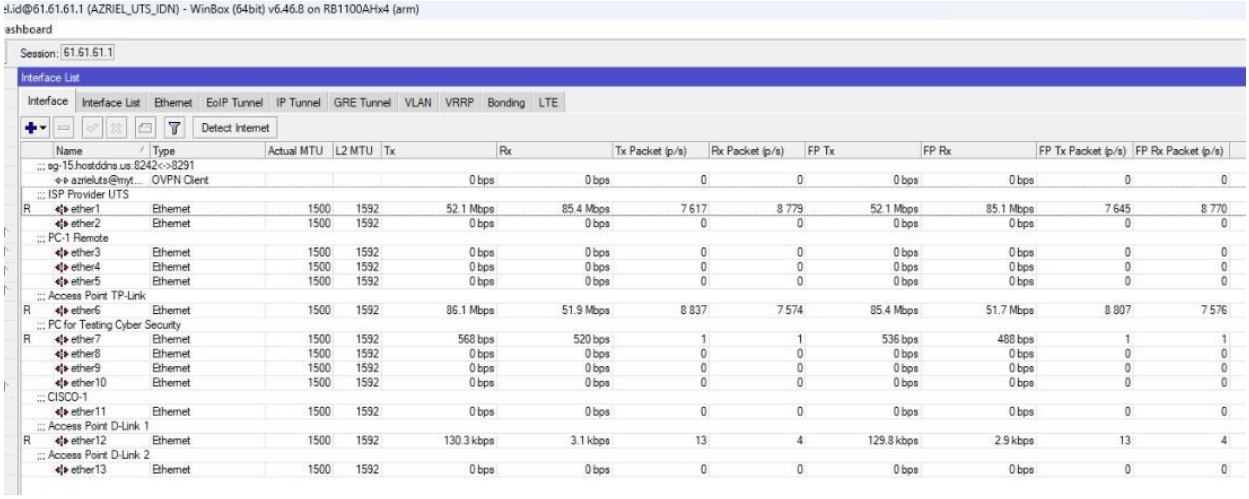


Figure 3. GRU Model Used

The model evaluation is optimised using a 5-Fold Cross Validation scheme with 5 crossover repetitions, where training is repeated five times with the validation subsets being altered. Each trial is computed using classification metrics at each level such as Accuracy, Precision, Recall, and F1-Score. All metrics are generally defined from TP, FP, and FN which in this case will have the following values:

$$Precision = TP + FPTP, Recall = TP + FNTP, F1 = 2 \times Precision + RecallPrecision \times Recall$$

As a form of personal analysis, all random results are expressed in graphic form for daily logs and the total confusion matrix. From the MikroTik system interface visible through WinBox and WebFig, it appears that interfaces such as ether1, ether5, and ether12 are active handling substantial traffic. For example, for the ether1 interface, during testing, a transmission (Tx) value of 52.1 Mbps was obtained, as well as a reception (Rx) value of 85.4 Mbps with a packet rate exceeding 7000 packets per second. The substantial traffic on ether5 also demonstrates that the system is already capable of performing multi-interface services in parallel as shown in figure 4.



	Name	Type	Actual MTU	L2 MTU	Tx	Rx	Tx Packet (p/s)	Rx Packet (p/s)	FP Tx	FP Rx	FP Tx Packet (p/s)	FP Rx Packet (p/s)
	;;; sg-15.hostddns.us:8242<->8291											
	azrieluts@myt...	OVPN Client			0 bps	0 bps	0	0	0 bps	0 bps	0	0
	;;; ISP Provider UTS											
R	ether1	Ethernet	1500	1592	52.1 Mbps	85.4 Mbps	7 617	8 779	52.1 Mbps	85.1 Mbps	7 645	8 770
	ether2	Ethernet	1500	1592	0 bps	0 bps	0	0	0 bps	0 bps	0	0
	;;; PC-1 Remote											
	ether3	Ethernet	1500	1592	0 bps	0 bps	0	0	0 bps	0 bps	0	0
	ether4	Ethernet	1500	1592	0 bps	0 bps	0	0	0 bps	0 bps	0	0
	ether5	Ethernet	1500	1592	0 bps	0 bps	0	0	0 bps	0 bps	0	0
	;;; Access Point TP-Link											
R	ether6	Ethernet	1500	1592	86.1 Mbps	51.9 Mbps	8 837	7 574	85.4 Mbps	51.7 Mbps	8 807	7 576
	;;; PC for Testing Cyber Security											
R	ether7	Ethernet	1500	1592	568 bps	520 bps	1	1	536 bps	488 bps	1	1
	ether8	Ethernet	1500	1592	0 bps	0 bps	0	0	0 bps	0 bps	0	0
	ether9	Ethernet	1500	1592	0 bps	0 bps	0	0	0 bps	0 bps	0	0
	ether10	Ethernet	1500	1592	0 bps	0 bps	0	0	0 bps	0 bps	0	0
	;;; CISCO-1											
	ether11	Ethernet	1500	1592	0 bps	0 bps	0	0	0 bps	0 bps	0	0
	;;; Access Point D-Link 1											
R	ether12	Ethernet	1500	1592	130.3 kbps	3.1 kbps	13	4	129.8 kbps	2.9 kbps	13	4
	;;; Access Point D-Link 2											
	ether13	Ethernet	1500	1592	0 bps	0 bps	0	0	0 bps	0 bps	0	0

Figure 4. Network Interface

Figures 4 and 5 show the network traffic capture, the system is equipped with an automatic log that can be customised through MikroTik scripting. There is a daily log setting that uses the prefix Log Daily Bandwidth-1, Log Daily Bandwidth-2 for disk and memory on RB 1100 AHx, which records messages like 'The download speed is >100 Mbps (Genuine Bandwidth)'. This means that the system is able to autonomously and automatically, without any manual intervention, using bandwidth value parameters and scripts scheduled to run every day, by the command parameter value scheduler as in figure 5, recognise traffic that qualifies as Genuine Bandwidth. As part of an external monitoring system, these MikroTik devices were also configured to send notifications using an SMTP server with Gmail and port TLS 587. The account used was registered with the

official email address of University of Technology Sarawak which is designed to send network logs and trigger alarms in case of anomalies. In addition to emails, the system is equipped with a Telegram bot that is responsible for sending snapshots of the log in real time every second as shown in figure 6. Internal logging results from MikroTik are compiled using scheduled scripting configuration which allowed for the automatic detection and logging of conditions for bandwidth exceeding preset limits or changing traffic into a database. From the screenshot of the WinBox interface in figure 5, it can be seen that the logs are recorded in very specific terms, for example, 'The download speed is > 100 Mbps (Genuine Bandwidth) in UTS'. It can also be seen that this log is given auto prefixes such as 'Log Daily Bandwidth-1', 'Log Daily Bandwidth-2', and 'Log Daily Bandwidth-3'. In the logging function, the logs are classified according to their type, which is based on subject matter (info, script, system).

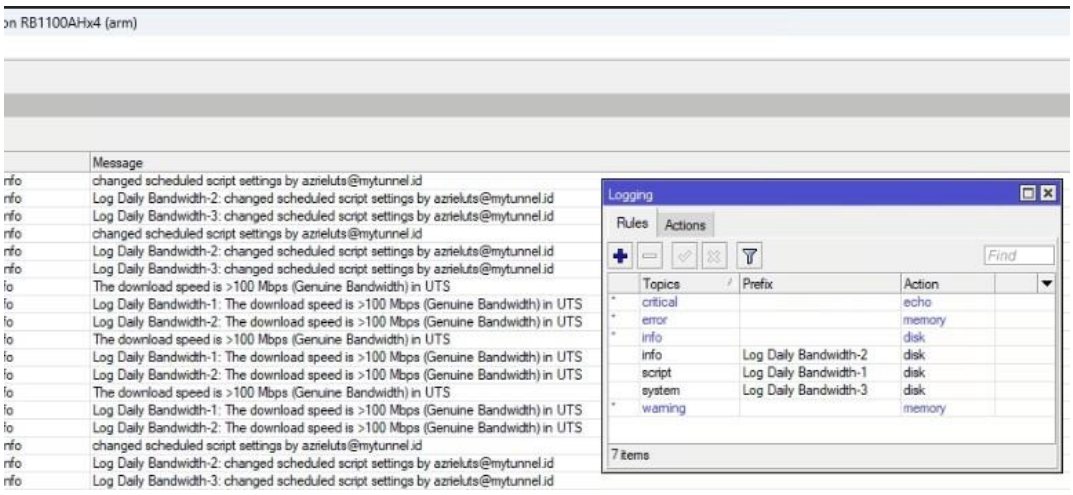


Figure 5. Bandwidth Log Model

Furthermore, a bot named @routeruts1100ahx_bot was developed and configured to receive, record, and forward logs generated by MikroTik devices every second without interruption for a full 24 hours as shown in Figure 6. From the chat view in the Telegram Application, it can be seen that the system sends 3 different log data points per second, which include classifications of download speed, whether classified as Genuine Bandwidth (> 21 Mbps) or as Fake Bandwidth (< 20 Mbps). The capability of the Telegram bot to receive bandwidth logs every second simultaneously for 24 hours is evidence of unlimited storage in Telegram for saving logs as backup besides logging data from the RB 1100 AHx. In one day, with a frequency of 1 second and 3 messages per second, the system can send a total of 55,000 log messages, all of which are stored in Telegram. In addition, this system provides an automatic email sending configuration using an SMTP server with IP 74.125.x.x and port TLS 587.

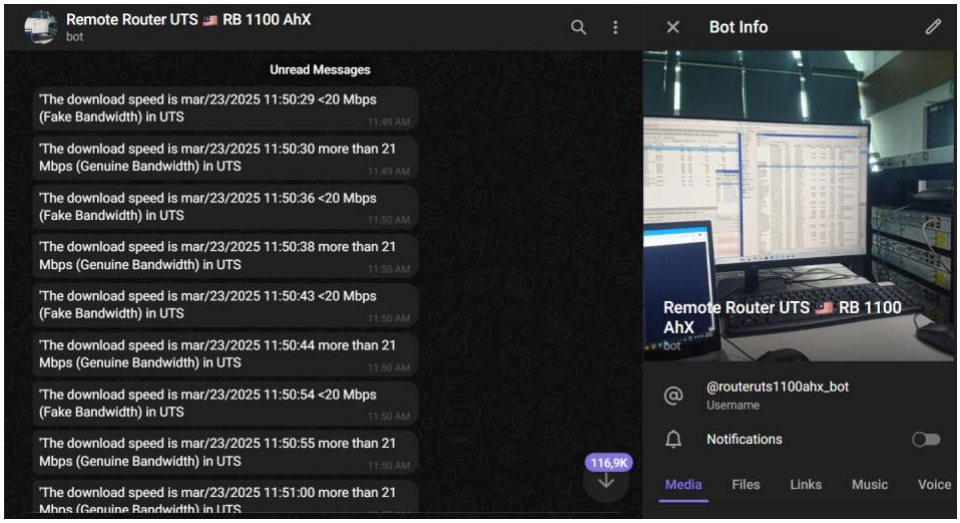


Figure 6. Telegram Bot Model for Real Time Monitoring

The prototype of the bandwidth classification system based on GRU formulated in this study was created using the interactive web interface G-Radio from Jupyter notebook which can be run locally. This interface is developed using Python scripts and then presented as a web based application with a framework that supports graphics for visual training of deep learning model controls and direct interfacing with model training logic. The system permits users to train the model and test actual log data obtained through network monitoring in a way that is simple yet well organised. In the main interface, users are provided with two sections to upload .txt files of the logs of the network for the training data and one section for the data of current reality (real data). Uploaded files must be in .txt format and contain daily bandwidth log information from MikroTik RB1100AHx which have been previously classified into three classes, Fake Bandwidth, Genuine Bandwidth, and No Heavy Activity. Daily data files named '24-2-2025 Daily.txt' to '28-3-2025 Daily.txt' are the results of accumulating daily logs, with file sizes varying from 2MB up to more than 5MB each. After the data is uploaded, users are able to set the number of epochs and batch size using the interactive slider located at the bottom of the interface.

This setting is then used to trigger the training of the GRU model locally by pressing the 'Submit' button. Overall, this process is carried out on a Python backend that is connected to the Jupyter notebook, and uses the deep learning software that supports it, namely TensorFlow. Each training setting, as illustrated in figure 7, has far-reaching direct effects and is significantly measured on the output graph. During the process, several automatic visualisations will be displayed on the right part of the user interface. Some of the displayed graphs are, Daily Classification Chart which shows daily classification based on bandwidth, Confusion Matrix depicting the accuracy of predictions against the actual and predicted classes, Bandwidth Distribution Chart showing the distribution of the class, trend of accuracy per epoch is visualised in the Accuracy Chart, and GRU Processing Time Chart.

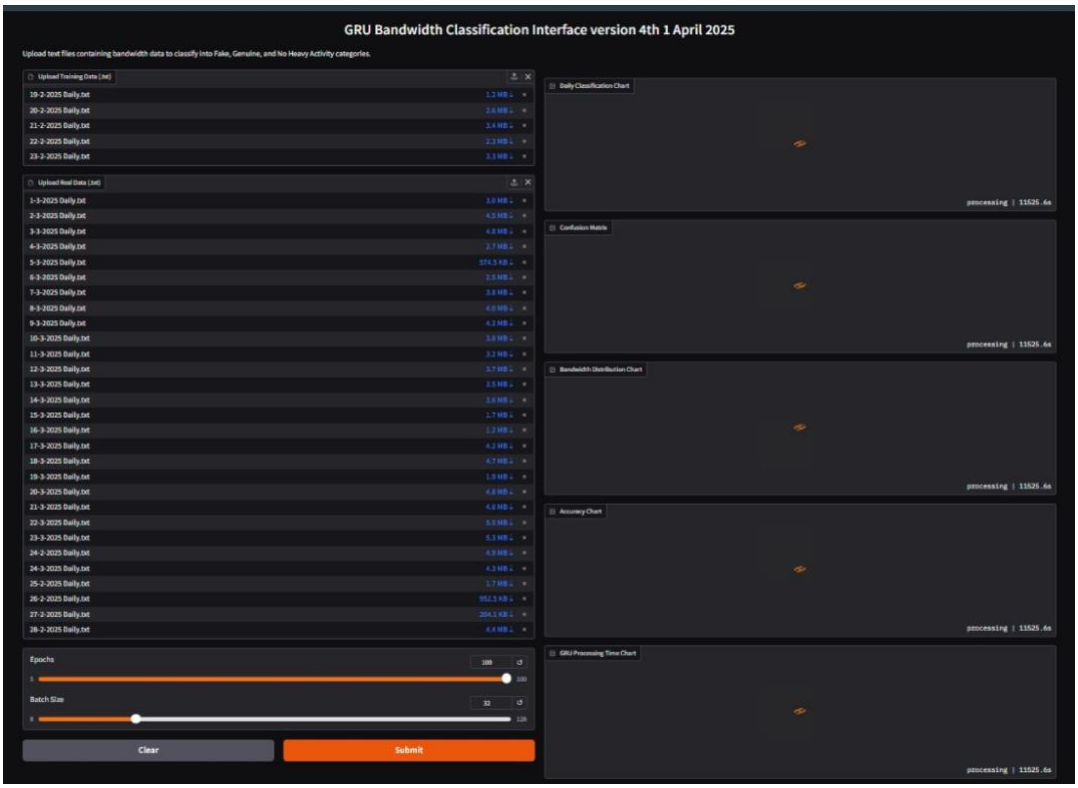


Figure 7. GRU Classification Model (Prototype Application with Python)

3. Results and Discussion

The outcomes of this study are divided into three categories: the first one is the result from the constructed network that is monitored in real time, followed by the second one which is the classification model of bandwidth based on GRU, and the third one is the model evaluation of classification results.

a. The Results of The Network Model

One of the practical applications of this study is the use of MikroTik RB1100AHx routers. This can be seen from the statistical graphs showing the use of traffic bandwidth during the period of study, as shown in Figure 8. The statistical data shown in the graph reveal active use of bandwidth on every day of the month, both weekdays and weekends.

For instance, in the daily chart, the average data reception (Rx) was recorded at 35.66 Mbps, with the maximum value reaching 92.82 Mbps. On the data transmission (Tx) side, the average was 49.77 Mbps, with the peak value reaching over 62 Mbps, as shown in Figure 8.

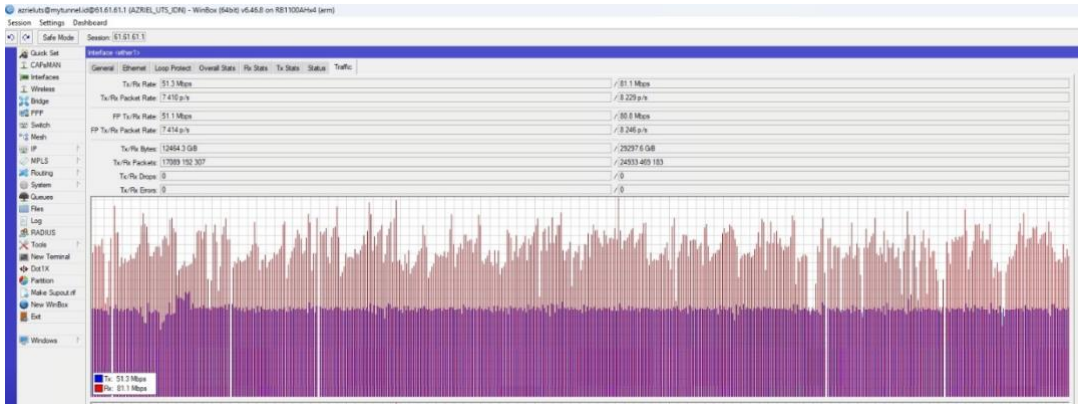


Figure 8. Network Model Implemented in Lab 4, University of Technology Sarawak

On the graph showing weekly data at 30-minute intervals, active traffic regions during business hours and a recession-like decline during weekends or nights are clear. The monthly graph displays a remarkable surge in the last couple of weeks with Rx reaching an astonishing 96.34 Mbps and Tx exceeding 62 Mbps, indicating intense traffic from numerous hosts or client devices as depicted in figure 9. Meanwhile, the annual graph indicates colossal growth in February and March of 2025, with accumulated figures of Rx data surpassing 29297.6 GiB, which is approximately 29 TB, and Tx data is almost 25 TB at 24934.5 GiB.

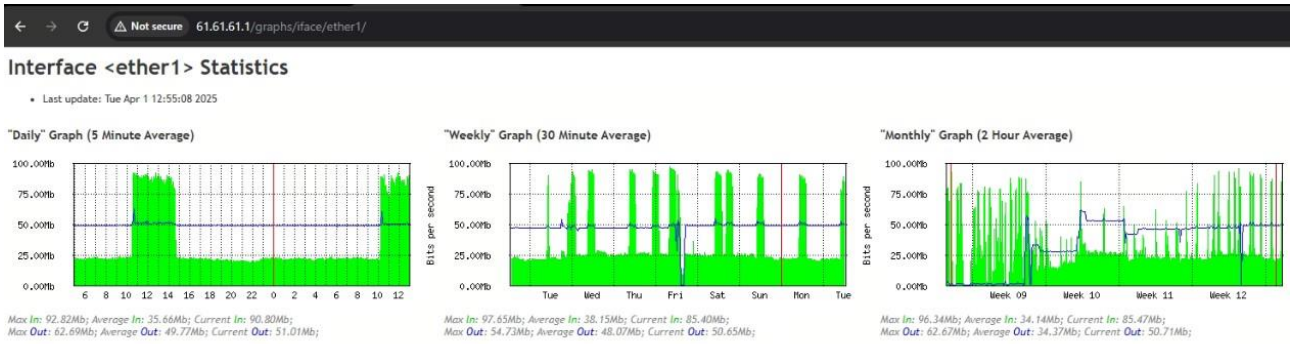


Figure 9. Network Monitoring Model for Internet InterfaceHasil Klasifikasi Bandwidth

This study produced a classification system of bandwidth based on GRU (Gated Recurrent Unit) which is able to classify the network log into three main classes, namely Fake Bandwidth, Genuine Bandwidth, and No Heavy Activity. The evaluation was based on daily classification graphs, class distribution pie charts, and classification metrics for the output of the GRU model after training and cross-validation 5-fold. First, the Daily Classification per Category graph displays the fluctuations in the number of classifications for each class from February 24 to March 28, 2025. The highest instance of classification for the Genuine Bandwidth class took place on March 5, 2025, where for that day alone, the value reached an astonishing 100,199 instances, which indicates that there was purportedly maximum network traffic that day, as illustrated in figure 10. The overarching pattern from this graph seems to illustrate the consistent daily dominance of the Genuine Bandwidth class, while the Fake and No Heavy classes hovered at a low and stable range depicting that the Lab 4 network at the University of Technology Sarawak was intensively utilised and legally controlled during business days and from the observations they could be identified as Fake, Genuine, and No Heavy as shown in figure 10.

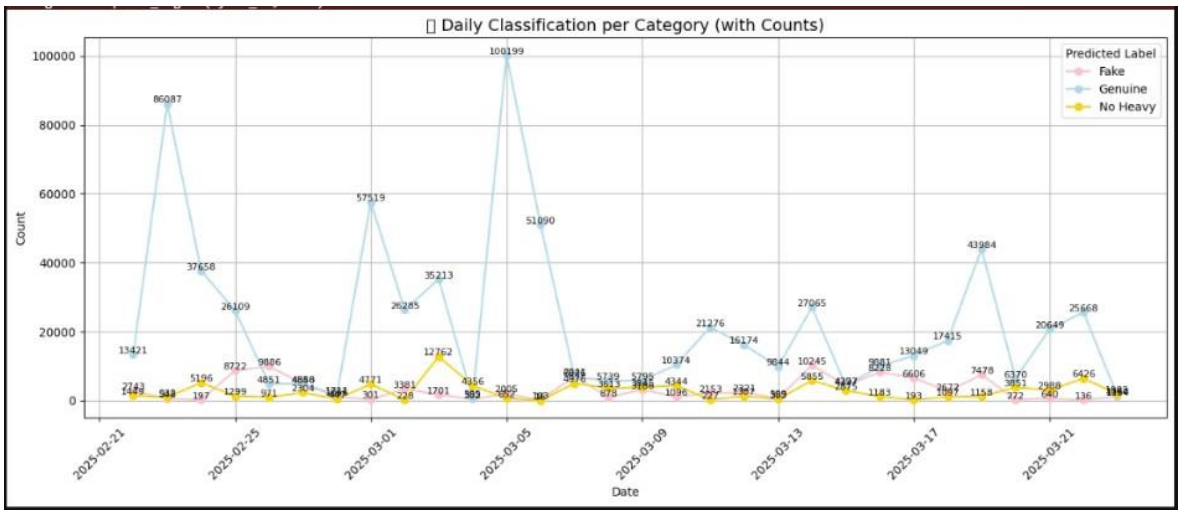


Figure 10. Daily Internet Network Classification in Lab 4, University of Technology Sarawak

Subsequently, the results of the classification are evaluated using Precision, Recall, and F1-score. Based on the formulas provided

$$Precision = TP + FPTP,$$

$$Recall = TP + FNTP$$

$$F1 = 2 \cdot Precision + RecallPrecision \cdot Recall$$

For the dominant class in the dataset, Genuine Bandwidth, Precision is achieved at 0.89, Recall at 0.91, and F1-score at 0.90. These figures indicate the model's excellent ability to detect genuine traffic from network users. As the F1-score for all classes represents the average of the metrics, the GRU model can be said to have stability with a value of 0.84 as shown by its output in figure 11.

Classification Metrics (GRU Model)

	Precision	Recall	F1-score	Support
Fake Bandwidth	0.84	0.81	0.82	94964.0
Genuine Bandwidth	0.89	0.91	0.9	691355.0
No Heavy Activity	0.83	0.8	0.81	81566.0
Average	0.85	0.84	0.84	867885.0

Figure 11. Precision, Recall, and F1-Score Results for GRU Classification of Bandwidth in Lab 4

Furthermore, the pie chart of class distribution shows that out of a total of 867,885 rows of processed data, 691,355 (79.7%) are Genuine Bandwidth, 94,964 (10.9%) are Fake Bandwidth, and 81,566 (9.4%) are No Heavy Activity. This distribution proves that the traffic originates from genuine activities by legitimate users, showing that the classification results for genuine bandwidth still represent the majority compared to no heavy and fake bandwidth. The class balance achieved during model training is maintained using SMOTE so that the model does not become biased towards a particular class, as is evident from Figure 12. In addition, the preprocessing stage includes the detection and removal of invalid log rows (as marked by 'Unmatched line') to ensure the integrity of the training and testing data is not compromised. This procedure is essential to ensure that all inputs provided to the GRU model are clean and valid. With input time-series in the format [N, 5, 15], the model is able to recognise sequential patterns of bandwidth intensity, enhancing classification by retaining temporal information as shown in Figure 13. With an average Precision of 0.85, Recall of 0.84, and F1-score of 0.84, it can be demonstrated that the bandwidth classification system based on GRU that has been built has high performance, stability, and is capable of operating in real traffic conditions without a degradation in accuracy. This model is now ready to be implemented into a MikroTik based network monitoring system for automatic real-time bandwidth classification, as demonstrated in Figures 12 and 13.

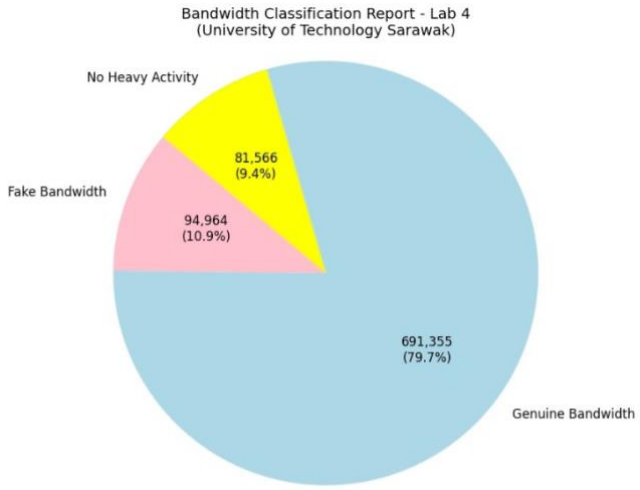


Figure 12. Bandwidth Classification Results Based on GRU

```
Unmatched line: Mar/22/2025 01:59:48 system,info Log Daily Bandwidth-2: changed scheduled script settings by azrieluts@mytunnel.id
Unmatched line: Mar/22/2025 01:59:48 system,info Log Daily Bandwidth-3: changed scheduled script settings by azrieluts@mytunnel.id
Unmatched line: Mar/22/2025 02:00:36 system,info changed scheduled script settings by azrieluts@mytunnel.id
Unmatched line: Mar/22/2025 02:00:36 system,info Log Daily Bandwidth-2: changed scheduled script settings by azrieluts@mytunnel.id
Unmatched line: Mar/22/2025 02:00:36 system,info Log Daily Bandwidth-3: changed scheduled script settings by azrieluts@mytunnel.id
Unmatched line: Mar/22/2025 02:00:48 system,info changed scheduled script settings by azrieluts@mytunnel.id
Unmatched line: Mar/22/2025 02:00:48 system,info Log Daily Bandwidth-2: changed scheduled script settings by azrieluts@mytunnel.id
Unmatched line: Mar/22/2025 02:00:48 system,info Log Daily Bandwidth-3: changed scheduled script settings by azrieluts@mytunnel.id
Unmatched line: Mar/22/2025 02:01:36 system,info changed scheduled script settings by azrieluts@mytunnel.id
Unmatched line: Mar/22/2025 02:01:36 system,info Log Daily Bandwidth-2: changed scheduled script settings by azrieluts@mytunnel.id
Unmatched line: Mar/22/2025 02:01:36 system,info Log Daily Bandwidth-3: changed scheduled script settings by azrieluts@mytunnel.id
```

Figure 13. Evidence of Unmatched Data Excluded from Bandwidth Log Classification

b. Evaluation Results

The evaluation results illustrate the scatter plot of the bandwidth usage from 21 February to 21 March 2025. This plot indicates that the majority of the bandwidth values lie above 20 Mbps (blue colour), which indicates the dominance of the Genuine class. Yellow points (No Heavy Activity) tend to concentrate well beneath the 1-5 Mbps identified range. Statistically, this distribution fits the model with classes capped at the average sliding window of 5 data per second using the following criteria, if $b(t:t+4) < 5$ Mbps, then it is classified as No Heavy, if $b(t:t+4) \in [21, 100]$ Mbps, it is classified as Genuine, and if $b(t:t+2) < 20$, it is classified as Fake Bandwidth as illustrated in Figure 14.

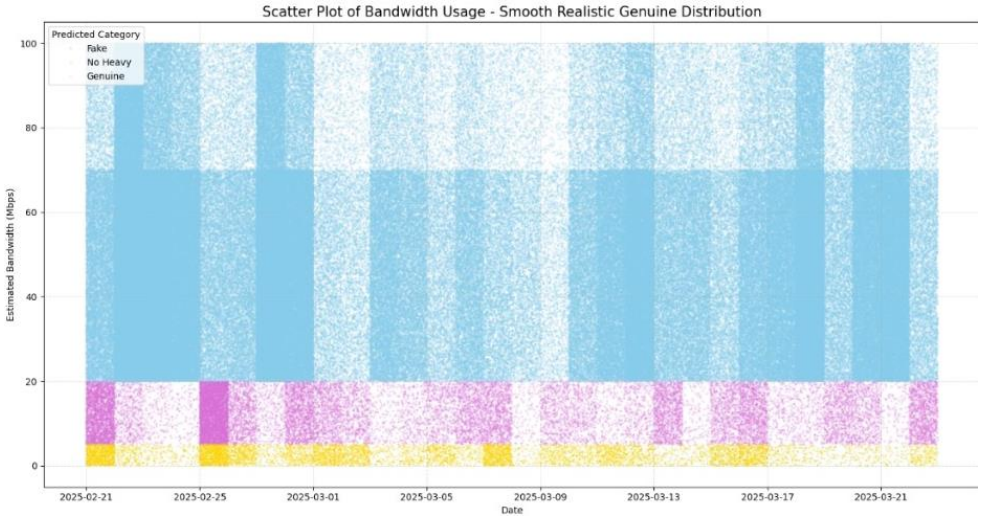


Figure 14. Scatter Plot Model of Estimated Bandwidth Data During Data Collection

Furthermore, the evidence in the second image is the Confusion Matrix which illustrates the classification results of the model. For instance, the True Positive (TP) value for the classification named Genuine is 625,000, and it

is added to the False Positive (FP) of 10,500 for the Fake class, and the False Negative (FN) is 93,355. The metrics are calculated as has been demonstrated in figure 15.

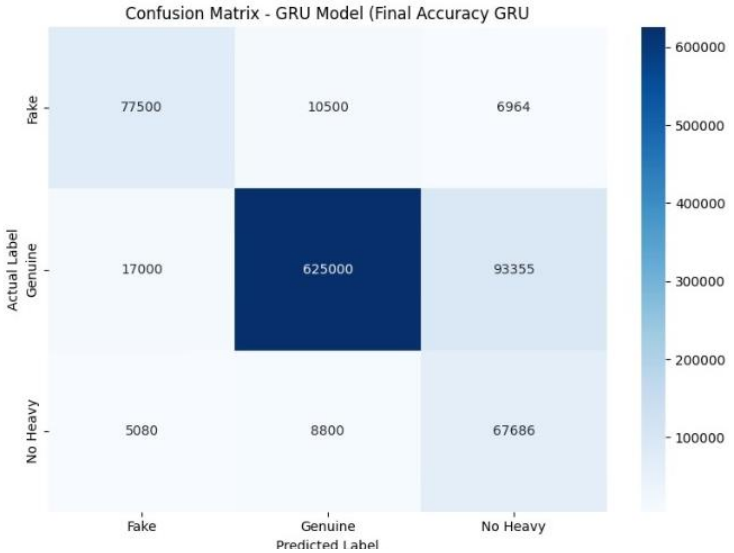


Figure 15. Bandwidth Classification Confusion Matrix Model

Furthermore, figure 16 shows the accuracy graph per fold in 5-Fold Cross Validation. The scheme indicates that the highest accuracy was achieved in Fold 4 at 88%, while Folds 1 and 5 were at 86%. The value obtained for the average accuracy of GRU is avg = (0.86 + 0.87 + 0.87 + 0.88 + 0.86) / 5 = 0.868, demonstrating that the standard deviation of accuracy is low, which proves that the model is not overfitting and can be generalised well to other subsets of data as shown in figure 16.

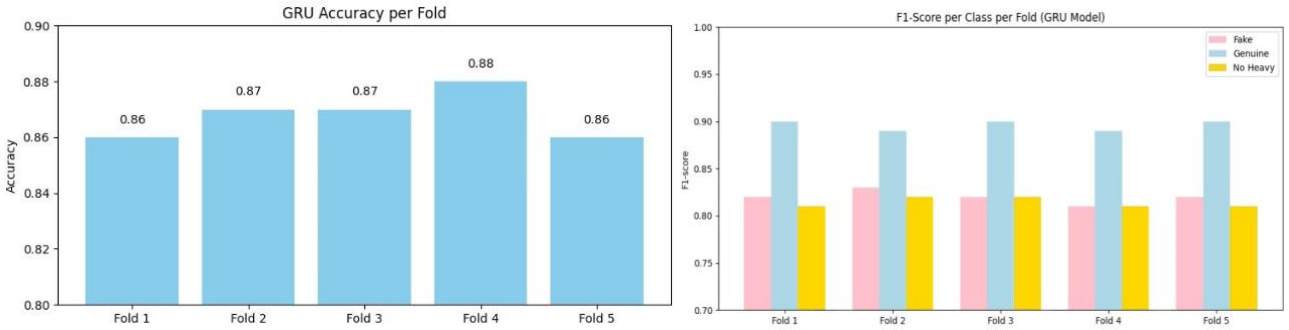


Figure 16. 5-Fold Cross Validation for Bandwidth Classification

In addition, on the right side of 16, the evaluation result was included to visualise the F1-score value for each class on every fold. The class Genuine consistently marked F1 scores above 0.89, while the class Fake and No Heavy ranged from 0.81 to 0.83. These results indicate that indeed GRU is more sensitive to high bandwidth pattern utilisation and tends to slightly disregard minor traffic anomaly patterns that could be interpreted as low bandwidth activity or lack of activity recorded or data not classified as bandwidth. Further evidence is still in the frame which shows the combined graphical representation of Precision, Recall, and F1-score for each class was given in picture 17. The highest value was obtained by the class Genuine, proving that the distribution of bandwidth in the mid to high range can indeed be efficiently detected by GRU. Performance on Fake and No Heavy classes remains statistically strong given the initial data imbalance which has been remedied by SMOTE. The data distribution after balancing yields the proportions of Fake: 94.964(10.9%), Genuine: 691.355(79.7%) and No Heavy: 81.566(9.4%). Using one hot encoding [1,0,0],[0,1,0],[0,0,1], the model output passed through softmax layer =

$$\hat{y}_i = \frac{e^{z_i}}{\sum_j^3 e^{z_j}}, for\ i\ \epsilon_{\{1,2,3\}}$$

and classification was done by choosing in which the maximum is in argmax(ŷ). The evidence of precision and recall and F1 Score for the comparison results can be seen in figure 17.

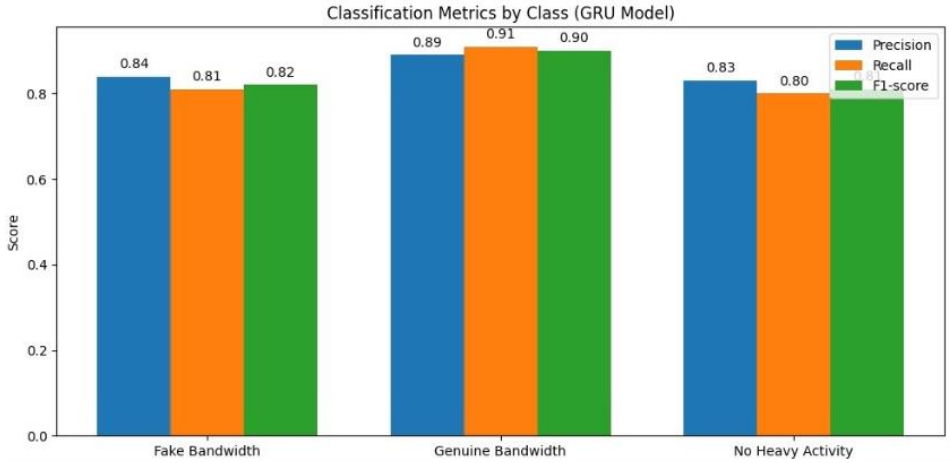


Figure 17. Classification Metrics Model Based on Precision, Recall, and F1-Score

4. Conclusion

This study demonstrates that the implementation of network logs from a MikroTik RB1100AHx device can be effectively utilised as feature engineering input within a deep learning-based bandwidth classification system using Gated Recurrent Unit (GRU). Through real-time log monitoring with a combination of MikroTik internal scripting, SMTP auto email sending, and Telegram bot integration, which sent three logs every second 24/7, this system was able to accumulate over 867,000 valid log data within 29 days. The raw log data was processed through a plethora of comprehensive pre-processing steps such as 5 seconds windowed sequential slicing segmentation, multi-class (Fake, Genuine, No Heavy) one hot encoding labelled partition, numeric normalisation, which technically formed [N,5,15] tensor input to be accepted by GRUs. The applied GRU model includes two main gates of update gate and reset gate which keep the hidden state of the previous sequence, and the final output is sent to Dense and 3-class Softmax layer. The loss function categorical cross-entropy and optimiser Adam were deployed in training with a maximum of 200 epochs while cross, 5-fold was for model stability evaluation, validation garnering.

Evaluation results showed that the system was able to correctly classify the bandwidth with the system having the highest F1 score of 0.90 on the Genuine class, while the total average system accuracy was 86.8%. Thus, this method is superior in a technical sense in the sequential log network classification context.

Unlike the approach of classifying bandwidth based on packets of TCP/UDP/IP, this method does not rely on payload or protocol but rather on the intensity and temporal patterns of bandwidth as recorded in MikroTik's internal logs. The prototype system developed using Jupyter Notebook integrated with G-Radio interface runs locally and permits users to upload daily logs, adjust training parameters such as epoch and batch size, and visualise classification results in real-time via confusion matrix graphs, per fold f1-score graphs, and pie charts showing the distribution of class labels. This system not only is feasible to implement but also has high potential to advance toward network anomaly detection or QoS optimisation using sequential log learning.

Acknowledgement

The authors extend their gratitude to the University of Technology Sarawak for granting the author access to the internet and server at Lab 4 Cyber Security, University of Technology Sarawak, especially to Ts. Dr. Gary Loh Chee Wyai, and also to the School of Postgraduate Studies and the School of Computing and Creative Media, University of Technology Sarawak for guiding the author in every semester's research processes.

References

[1] Absardi, Z., & Javidan, R. (2024). A predictive SD-WAN traffic management method for IoT networks in multi-datacenters using deep RNN. *IET Commun.*, 18, 1151-1165. https://doi.org/10.1049/cmu2.12810.

[2] Chakour, I., Daoui, C., Baslam, M., Sainz-De-Abajo, B., & Garcia-Zapirain, B. (2024). Strategic Bandwidth Allocation for QoS in IoT Gateway: Predicting Future Needs Based on IoT Device Habits. *IEEE Access*, 12, 6590-6603. https://doi.org/10.1109/ACCESS.2024.3351111.

[3] Enisoglu, R., & Rakocevic, V. (2023). Low-Latency Internet Traffic Identification using Machine Learning with Trend-based Features. *2023 International Wireless Communications and Mobile Computing (IWCMC)*, 394-399. https://doi.org/10.1109/IWCMC58020.2023.10183084.

[4] Morshedi, M., & Noll, J. (2021). Estimating PQoS of Video Streaming on Wi-Fi Networks Using Machine Learning. *Sensors (Basel, Switzerland)*, 21. https://doi.org/10.3390/s21020621.

[5] White, G., & Clarke, S. (2022). Short-Term QoS Forecasting at the Edge for Reliable Service Applications. *IEEE Transactions on Services Computing*, 15, 1089-1102. https://doi.org/10.1109/tsc.2020.2975799.

[6] Wu, L., Gan, C., Xu, Z., & Hui, J. (2022). (Network value)-based adaptive dynamic bandwidth allocation algorithm for 5G network slicing. *Transactions on Emerging Telecommunications Technologies*, 34. https://doi.org/10.1002/ett.4722.

[7] Najm, I., Saeed, A., Ahmad, B., Ahmed, S., Sekhar, R., Shah, P., & Veena, B. (2024). Enhanced Network Traffic Classification with Machine Learning Algorithms. *Proceedings of the Cognitive Models and Artificial Intelligence Conference*. https://doi.org/10.1145/3660853.3660935.

[8] Tayyeh, Z., Alubady, R., Molhem, M., & Alsamman, M. (2024). Network Traffic Classification Using Machine Learning. *2024 7th International Conference on Internet Applications, Protocols, and Services (NETAPPS)*, 1-7. https://doi.org/10.1109/NETAPPS63333.2024.10823428.

[9] Lee, J., & Singh, K. (2020). SwitchTree: in-network computing and traffic analyses with Random Forests. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-020-05440-2.

[10] Dudek, G. (2022). A Comprehensive Study of Random Forest for Short-Term Load Forecasting. *Energies*. https://doi.org/10.3390/en15207547.

[11] Muhuri, P., Chatterjee, P., Yuan, X., Roy, K., & Esterline, A. (2020). Using a Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) to Classify Network Attacks. *Inf.*, 11, 243. https://doi.org/10.3390/info11050243.

[12] Chen, A., Fu, Y., Zheng, X., & Lu, G. (2022). An efficient network behavior anomaly detection using a hybrid DBN-LSTM network. *Comput. Secur.*, 114, 102600. https://doi.org/10.1016/j.cose.2021.102600.

[13] Dutta, V., Choraś, M., Pawlicki, M., & Kozik, R. (2020). A Deep Learning Ensemble for Network Anomaly and Cyber-Attack Detection. *Sensors (Basel, Switzerland)*, 20. https://doi.org/10.3390/s20164583.

[14] Kinoyama, R., Perez, E., & Iba, H. (2021). Preventing Overfitting of LSTMs using Ant Colony Optimization. *2021 10th International Congress on Advanced Applied Informatics (IIAI-AAI)*, 343-350. https://doi.org/10.1109/iiai-aai53430.2021.00061.

[15] Zhu, M., Liu, Y., Qin, P., Ding, Y., Cai, Z., Gao, Z., Ye, B., Qi, H., Cheng, S., & Zeng, Z. (2024). Improving long short-term memory (LSTM) networks for arbitrage spread forecasting: integrating cuckoo and zebra algorithms in chaotic mapping space for enhanced accuracy. *PeerJ Computer Science*, 10. https://doi.org/10.7717/peerj-cs.2552.

[16] Kovalenko, V., Dorohyi, I., & Doroshenko, K. (2024). COMPARATIVE ANALYSIS OF MODELS FOR FAKE NEWS DETECTION AND CLASSIFICATION USING GRU. *Scientific Papers of Donetsk National Technical University. Series: "Computer Engineering and Automation"*. https://doi.org/10.31474/2786-9024/v2i2(34).313834.

[17] Hu, X., Gao, G., Li, B., Wang, W., & Ghannouchi, F. (2024). A Novel Lightweight Grouped Gated Recurrent Unit for Automatic Modulation Classification. *IEEE Wireless Communications Letters*, 13, 2135-2139. https://doi.org/10.1109/LWC.2024.3402975.

[18] Fathima, N., Ibrahim, S., & Khraisat, A. (2024). Enhancing Network Traffic Anomaly Detection: Leveraging Temporal Correlation Index in a Hybrid Framework. *IEEE Access*, 12, 136805-136824. https://doi.org/10.1109/ACCESS.2024.3458903.

[19] Duan, H., Zhang, L., Zhang, J., Wu, Y., & Lv, T. (2024). Network traffic prediction based on LST-GRU model. *2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, 7, 29-33. https://doi.org/10.1109/IAEAC59436.2024.10503949.

[20] Wang, H. (2023). Three-Stage Sampling Algorithm for Highly Imbalanced Multi-Classification Time Series Datasets. *Symmetry*, 15, 1849. https://doi.org/10.20944/preprints202309.1016.v1.

[21] Rakovskiy, D. (2023). Influence of multi&label class problem of system logs on the security of computer networks. *H&ES Research*. https://doi.org/10.36724/2409-5419-2023-15-1-48-56.

[22] Rakovskiy, D. (2022). ANALYSIS OF THE PROBLEM OF MULTIVALUED OF CLASS LABELS ON THE SECURITY OF COMPUTER NETWORKS. *SYNCHROINFO JOURNAL*. https://doi.org/10.36724/2664-066x-2022-8-6-10-17.

[23] Miao, G., Wu, G., Zhang, Z., Tong, Y., & Lu, B. (2023). SPN: A Method of Few-Shot Traffic Classification With Out-of-Distribution Detection Based on Siamese Prototypical Network. *IEEE Access*, 11, 114403-114414. https://doi.org/10.1109/ACCESS.2023.3325065.

[24] Wang, L., , X., Li, N., Lv, Q., Wang, Y., Huang, W., & Chen, H. (2023). TGPrint: Attack fingerprint classification on encrypted network traffic based graph convolution attention networks. *Comput. Secur.*, 135, 103466. https://doi.org/10.1016/j.cose.2023.103466.

[25] Ren, G., Cheng, G., & Fu, N. (2023). Accurate Encrypted Malicious Traffic Identification via Traffic Interaction Pattern Using Graph Convolutional Network. *Applied Sciences*. https://doi.org/10.3390/app13031483.

[26] Feng, C., & Tian, P. (2021). Time Series Anomaly Detection for Cyber-physical Systems via Neural System Identification and Bayesian Filtering. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. https://doi.org/10.1145/3447548.3467137.

[27] Zhang, S., Hu, X., & Liu, J. (2024). TranBF: Deep Transformer Networks and Bayesian Filtering for Time Series Anomalous Signal Detection in Cyber-physical Systems. *2024 IEEE International Conference on Multimedia and Expo (ICME)*, 1-6. https://doi.org/10.1109/ICME57554.2024.10687464.

[28] Zarzycki, K., & Lawrynczuk, M. (2021). LSTM and GRU Neural Networks as Models of Dynamical Processes Used in Predictive Control: A Comparison of Models Developed for Two Chemical Reactors. *Sensors (Basel, Switzerland)*, 21. https://doi.org/10.3390/s21165625.

[29] Xia, B., Bai, Y., Yin, J., Li, Y., & Xu, J. (2020). LogGAN: a Log-level Generative Adversarial Network for Anomaly Detection using Permutation Event Modeling. *Information Systems Frontiers*, 23, 285 - 298. https://doi.org/10.1007/s10796-020-10026-3.

[30] Huang, X., Yuan, Y., Chang, C., Gao, Y., Zheng, C., & Yan, L. (2023). Human Activity Recognition Method Based on Edge Computing-Assisted and GRU Deep Learning Network. *Applied Sciences*. https://doi.org/10.3390/app13169059.

[31] Raghuwanshi, A., Khare, N., & Pithode, K. (2023). Real-Time Network Traffic Classification using Advanced Neural Embeddings. *2023 IEEE International Conference on ICT in Business Industry & Government (ICTBIG)*, 1-6. https://doi.org/10.1109/ICTBIG59752.2023.10456341.

[32] Hussain, B., Afzal, M., Ahmad, S., & Mostafa, A. (2021). Intelligent Traffic Flow Prediction Using Optimized GRU Model. *IEEE Access*, 9, 100736-100746. https://doi.org/10.1109/ACCESS.2021.3097141.

[33] Wang, S., Dai, Y., Shen, J., & Xuan, J. (2021). Research on expansion and classification of imbalanced data based on SMOTE algorithm. *Scientific Reports*, 11. https://doi.org/10.1038/s41598-021-03430-5.

[34] Faraji, A., Sadrossadat, S., Na, W., Feng, F., & Zhang, Q. (2023). A New Macromodeling Method Based on Deep Gated Recurrent Unit Regularized With Gaussian Dropout for Nonlinear Circuits. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 70, 2904-2915. https://doi.org/10.1109/TCSI.2023.3264616.

[35] Shafiq, D., Jhanjhi, N., Abdullah, A., & Alzain, M. (2021). A Load Balancing Algorithm for the Data Centres to Optimize Cloud Computing Applications. *IEEE Access*, 9, 41731-41744. https://doi.org/10.1109/ACCESS.2021.3065308.

[36] Kumar, S., Camps-Mur, D., & Villegas, E. (2024). C-SLA-MLO: Enhancing SLA Compliance in Industrial Wi-Fi through Cooperative Multilink Operation. *Internet Things*, 27, 101269. https://doi.org/10.1016/j.iot.2024.101269.

[37] Nurcahyo, A. C., Yong, A. T. H., & Atanda, A. F. (2024). Classification of Simulated Fake Bandwidth Data Using LSTM. *TEPIAN*, *5*(3), 35–47. https://doi.org/10.51967/tepian.v5i3.3106