MyJICT
Malaysian Journal of Information and Communication Technology
EISSN 0127-7812

Vol. 11, Issue 1, pp. 109-128 | JUNE 2026
DOI: https://doi.org/10.53840/myjict11-1-250

COMPUTER NETWORK MODEL AND ITS CLASSIFICATION BASED ON LSTM FOR IDENTIFYING FAKE BANDWIDTH

[i*]Azriel Christian Nurcahyo, [ii]Yiiong Siew Ping, & [iii] Huong Yong Ting

[1,2,3]School of Computing & Creative Media, University of Technology Sarawak, Sibu, Sarawak, Malaysia

*(Corresponding author) e-mail: pic24030001@student.uts.edu.my

Article history:
Submission date: 9 March 2026
Received in revised form: 22 May 2026
Acceptance date: 30 May 2026
Available online: 30 June 2026

Keywords:
Internet Service Provider, Long Short-Term Memory, Fake Bandwidth, Service Level Agreement

Funding:
This research was funded by a scholarship from Shanti Bhuana Institute, Indonesia and also supported with equipment assistance from University of Technology Sarawak, Malaysia

Competing interest:
The author(s) have declared that no competing interests exist.

Cite as:
Nurcahyo, A. C., Ping, Y. S., & Ting, H. Y. (2026). Computer Network Model and Its Classification Based on LSTM For Identifying Fake Bandwidth. *Malaysian Journal of Information and Communication Technology (MyJICT), 11(1), 109-128.* https://myjict.uis.edu.my/index.php/journal/article/view/250

ABSTRACT

The phenomenon of fake bandwidth arises when the actual internet throughput delivered by an Internet Service Provider consistently falls below the speed guaranteed in a Service Level Agreement. This study presents a Long Short-Term Memory classification framework to identify genuine and fake bandwidth based on real-world network log data collected at the University of Technology Sarawak, Malaysia. The network model was developed following the infrastructure pattern of the campus dedicated bandwidth environment connected to the Cyber Security Laboratory. A dataset comprising 1,857,285 log records was acquired over 60 days of continuous monitoring via a RB1100AHx backbone router integrated with a Telegram Bot based real time notification system. Four bandwidth categories were defined in accordance with ETSI quality standards, Genuine Bandwidth ≥21 Mbps, Fake Bandwidth <15 Mbps, No Heavy Activity <100 kbps, and Unclassified. The LSTM model was optimised through a 100-dimensional embedding layer, a Bidirectional LSTM layer with 128 units and progressive dropout regularisation (0.4–0.6), and the Adam optimiser at a learning rate of 0.0005. Experiments were conducted under three data split configurations (30:70, 50:50, and 70:30). The model achieved a consistent classification accuracy of 96.93% across all configurations, with an empirical error rate of 3.06%–3.07% and a theoretical generalisation bound of 2.87% derived from Vapnik–Chervonenkis theory. K-fold cross validation yielded a mean accuracy of 0.940, confirming model stability. Empirical analysis reveals that approximately 52% of monitored dedicated bandwidth constitutes genuine throughput, while approximately 23% exhibits fake bandwidth behaviour, providing quantitative evidence of systematic SLA non-compliance.

Introduction

In today's highly digital era, computer network infrastructure remains the backbone supporting nearly every aspect of modern digital life, ranging from personal communication and business operations to educational services (El-Hajj, 2025; Serrano, 2023; Hasan, Alzuod, et al., 2025). Quality of Service (QoS) provided by computer networks is a fundamental factor in determining the overall quality of internet performance in supporting various online service activities. (QoS (Quality of Service) Analysis on Internet Network," 2019; Alfharizi et al., 2026). At present, millions of internet users access the network each day with the expectation of receiving bandwidth in accordance with the Service Level Agreement (SLA) agreed with their Internet Service Provider (ISP) (Robitza et al., 2017; United Nations Conference on Trade and Development [UNCTAD], 2021; Ofcom, 2023). However, the phenomenon of fake bandwidth has become a serious issue that affects users' trust in ISP services, as speed test results often display high figures while the actual speed experienced by users is significantly lower than what was promised (Nurcahyo, Huong Yong Ting, & Atanda, 2025; Nurcahyo, Yong, & Atanda, 2024; Nurcahyo, Yong, & Atanda, 2025). The issue of fake bandwidth occurs systematically, whereby the bandwidth model offered in dedicated services does not correspond to the expected performance, despite requiring substantially higher costs (Nurcahyo, Yong, & Atanda, 2024; Nurcahyo, Yong, & Atanda, 2025). Even in shared bandwidth services, there remains a discrepancy between the promised bandwidth stability and the actual performance experienced by users, including connection disruptions, high latency, and substantial packet loss (Prasad, 2016; Ahmed et al., 2023). Fake bandwidth occurs when an ISP claims to provide a certain level of speed, but the actual speed experienced by users is significantly lower, particularly during peak hours when it may fall to half or even one third of the promised rate (Nurcahyo, Huong Yong Ting, & Atanda, 2025; Nurcahyo, Yong, & Atanda, 2024). Evidence of fraudulent practices in the sale of bandwidth services is further reinforced by the increasing number of bandwidth corruption cases, including those related to the procurement of Base Transceiver Stations (BTS), in which bandwidth services are sold through substantial price mark ups of up to two or three times the original cost, yet still fail to deliver the promised network speed (Sabani et al., 2019; Ahmad et al., 2025). This situation represents an issue that must be addressed, as it disadvantages internet users who pay for the service on a monthly basis.As a university that relies heavily on information technology infrastructure to support various digital services, including e-learning, academic information systems, and digital library services, the University of Technology Sarawak requires high quality and consistently reliable dedicated bandwidth. Although the university has subscribed to a dedicated bandwidth service from Telekom Malaysia, concerns remain regarding whether the bandwidth delivered truly complies with the agreed Service Level Agreement (SLA). This uncertainty has led to the idea of developing an automated classification system capable of distinguishing between genuine and fake bandwidth based on actual network log data.

Literature Review

Previous studies in the field of computer network QoS classification have largely relied on network simulations using software such as GNS3, Cisco Packet Tracer, or NS3 (Nedyalkov, 2023; Fauzan et al., 2026; Nurfitri Handayani et al., 2024), which has not yet been able to accurately reflect network conditions in large scale environments, such as institutional networks (Gil et al., 2014). Carlos Güemes identified a significant gap between simulation-based approaches and data collected directly from real network systems. As a result, the classification outcomes tend to be less accurate when applied to real-world network scenarios (Güemes-Palau et al., 2025). Hassan Keshavarz and Ruaa Hasan emphasised the importance of using authentic data derived from operational networks, yet acknowledged that collecting such data is highly challenging (Keshavarz et al., 2021; Hasan, Kamal, et al., 2024). Edozie revealed that deep learning models trained on simulated data often experience a significant decline in performance when applied to real network traffic (Edozie et al., 2025). The most significant research gap lies in the limited number of studies that utilise large scale real network log data to train deep learning models, particularly Long Short-Term Memory (LSTM). Such approaches are difficult to implement due to the extensive processing time required (Yilmaz & Büyüktahtakın, 2023). Research conducted by Marijana Pavlov, Jordan D Chambers, and Ivan Malashin has demonstrated the considerable potential of Long Short-Term Memory (LSTM) in sequential data analysis however, further optimisation is still required (Pavlov-Kagadejev et al., 2024; Chambers et al., 2024; Malashin et al., 2024). Although LSTM is

designed to address the vanishing gradient problem and to learn long-term dependencies, its application to bandwidth classification still faces challenges in handling the complexity of real network data (Sinha et al., 2025). Dash identified that further research is required to investigate the optimisation of LSTM for computer network applications, particularly in relation to service optimisation (Dash et al., 2025). This limitation is largely due to the fact that sufficiently large training datasets required to train LSTM models effectively are often unavailable because access to such data is limited (Talaei Khoei et al., 2023). Pejman noted that the practical implementation of LSTM is hindered by the difficulty of obtaining and processing computer network log data, which often has not undergone proper data cleansing (Peykani et al., 2025). Johnson revealed that the effectiveness of deep learning in classifying data within computer networks still requires further optimisation, particularly in addressing the issue of class imbalance (Johnson & Khoshgoftaar, 2019). Javed identified that limitations in the quantity of training data may affect the model's performance and generalisation, thereby necessitating the use of private data that are more accurate and clearly verifiable (Javed et al., 2025). Tam emphasised the importance of enhancing real-time data parameters, including daily log data and other QoS factors such as the amount of bandwidth received (Tam et al., 2023). Nieminen emphasised that the use of real world data from production environments is essential for developing a robust model that is capable of adapting to contemporary conditions (Nieminen et al., 2026).

This study aims to address the identified research gap by implementing an optimised LSTM model for bandwidth classification based on integrated daily network log data derived from the network infrastructure developed at UTS Malaysia. Unlike previous studies that relied on simulations or limited public datasets, this research utilises an extensive dataset comprising more than 1.8 million bandwidth log records collected over 60 days of continuous monitoring from UTS's dedicated bandwidth network, operating at 200-300 Mbps in accordance with the SLA agreement with Telekom Malaysia and connected to Lab 4 Cyber Security. The dataset was collected in real time using a MikroTik Router RB1100AHx configured with a logging system that was fully integrated, generating between 20,000 and 55,000 log entries per day in .txt format. The dataset includes four classification categories based on ETSI (European Telecommunications Standards Institute) standards, adapted and simplified for this study: Genuine Bandwidth (≥21 Mbps), Fake Bandwidth (<15 Mbps), No Heavy Activity (<100 kbps), and Unclassified, given the 200 Mbps dedicated bandwidth configuration. The novelty of this research lies in five principal aspects. First, the implementation of a real time bandwidth log data acquisition system using the MikroTik RB1100AHx equipped with 13 Ethernet interfaces, a firewall comprising 43 security rule models, and integration with Telegram Bot and email notifications for continuous daily monitoring. Second, the optimisation of the LSTM architecture through a Bidirectional LSTM layer with 128 units, progressive dropout regularisation (0.4–0.6), and the Adam optimiser with a learning rate of 0.0005, deployed across two servers for model execution. Third, the implementation of a structured training strategy using callback mechanisms, including EarlyStopping, ModelCheckpoint, and ReduceLROnPlateau, to enhance model performance and stability. Fourth, a systematic experimental design employing three data split ratios (30:70, 50:50, and 70:30) to evaluate model stability across different training testing distributions. Fifth, the application of Vapnik Chervonenkis theory to compute a dynamic error rate with a 95% confidence level, thereby improving the reliability of the reported accuracy and reducing error rates. Beyond these five novelty aspects, the present study is further distinguished by the scale and authenticity of its dataset. Unlike prior works that employed GNS3 simulations (Nedyalkov, 2023), Cisco Packet Tracer environments (Fauzan et al., 2026), or NS3 emulations (Nurfitri Handayani et al., 2024) none of which can fully replicate the stochastic nature of live institutional traffic this research utilises 1,85 milion real network log records continuously collected over 60 days from the University of Technology Sarawak operational network. This dataset scale and real-world provenance constitute a methodological advancement over benchmark datasets commonly used in deep learning-based network classification literature (Edozie et al., 2025). Furthermore, when contextualised against the same research group's prior architectural studies specifically the GRU based bandwidth classification achieving approximately 95.2% accuracy (Nurcahyo, Huong Yong Ting, & Atanda, 2025) and the CNN based classification achieving approximately 95.8% (Nurcahyo, Yong, & Atanda, 2025) the bidirectional LSTM proposed in this study achieves a higher and more stable accuracy of 96.93% across all three data split configurations, confirming the superiority of recurrent memory cell architectures for large scale temporal network log classification. The objectives of this study are to implement a computer network system capable of recording daily network logs from the UTS Malaysia network architecture, producing a dataset exceeding

1.8 million classified samples; to optimise the LSTM model for genuine and fake bandwidth classification using training testing splits of 30%, 50%, and 70%, alongside K-Fold cross-validation; and finally, to evaluate model performance using accuracy, K-Fold cross-validation, recall, F1-score, training accuracy, validation accuracy, training loss, validation loss, and processing time for each split variation, executed on independent servers.The significance of this research lies in its potential long-term impact for network administrators who depend on bandwidth services, as well as for researchers in the networking domain, particularly within ISPs. The study aims to provide empirical evidence that user measured bandwidth can be more accurately classified to determine its authenticity. The expected outcomes include an LSTM based bandwidth classification framework with high accuracy (>96%), deeper insight into LSTM behaviour on large-scale real-world network datasets, and a performance benchmark across three data split variations.

Methodology

Network Architecture and System Configuration

The network infrastructure was implemented using RouterOS version 6.46.8 on a MikroTik RB1100x4 device with the system identity AZRIEL_UTS. The system was configured with the Asia/Kuala_Lumpur time zone and an NTP client to ensure accurate time synchronisation, which is essential for recording real-time log data on a daily basis. Logging was configured to store up to 55,000 lines in the router's disk file to ensure that no critical data was lost. The configuration involved 13 Ethernet interfaces; each assigned a specific function. Ether1 served as the main connection to the UTS ISP Provider (dedicated bandwidth of 200–300 Mbps) using a DHCP client. Ether2 and Ether3 (30.30.30.1/24) were configured for remote management. Ether6 (61.61.61.1/24) was connected to a TP-Link access point to provide wireless connectivity. Ether7 (70.70.70.1/24) was used for cybersecurity testing involving several computers with open access. Furthermore, Ether11 (11.11.11.1/24) was connected to the core switch and distributed to open access points in order to replicate the Extended Service Set (ESS) network model used in the existing campus network. Meanwhile, Ether12 (120.120.120.1/24) and Ether13 (192.168.13.1/24) were configured for wireless network expansion through TP-LINK MR3420 access points, as illustrated in Figure 1.

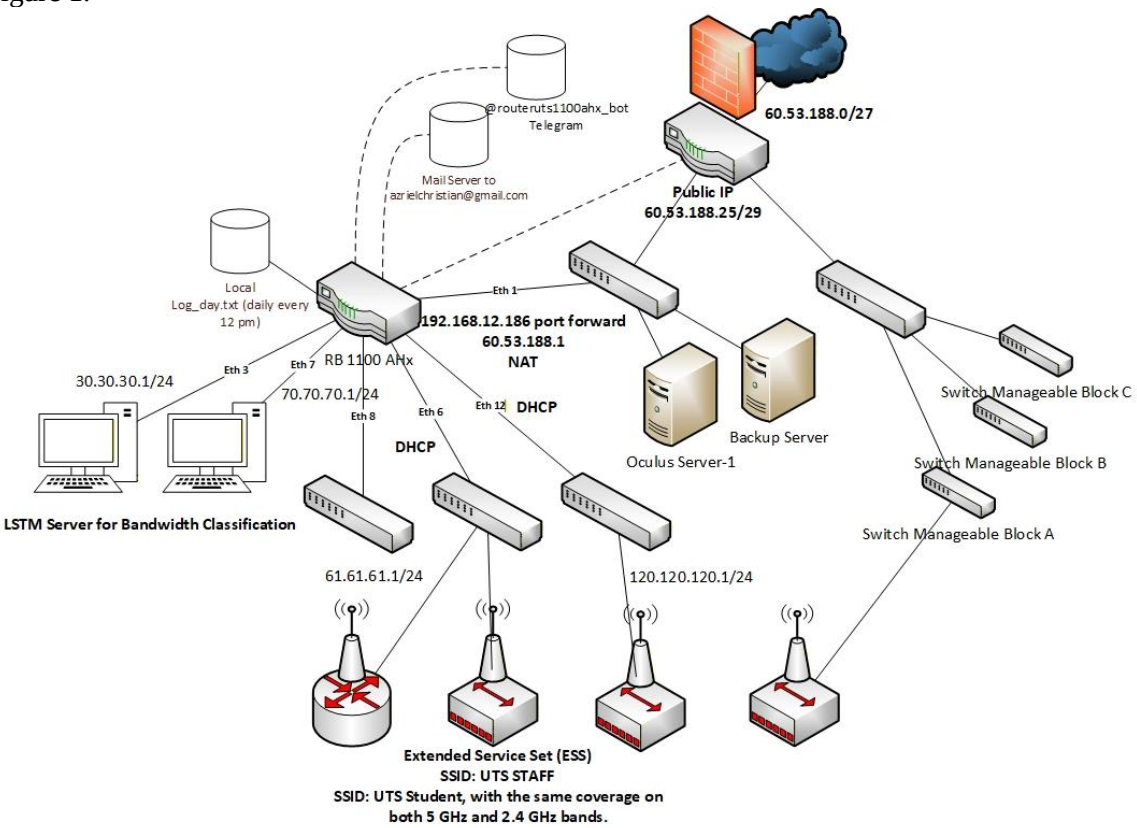

Figure 1: Network Model at the University of Technology Sarawak

Network Security System Model

The network security system was implemented through a virus filtering mechanism within the firewall, consisting of 43 comprehensive rules designed to block various types of malicious traffic. These rules specifically target known worms such as Blaster, Messenger, and Sasser, backdoors including MyDoom, NetBus, and SubSeven and trojans such as OptixPro, Trinoo, and PhatBot. In addition, several commonly exploited ports (135–139, 445, and 4444) were blocked to prevent unauthorised access and potential attacks on the network.Quality of Service (QoS) for traffic management was implemented using Simple Queue configuration. The system defined LAN 7 with a bandwidth limit of 1G/1G and LAN 12 with a limit of 3.5G/3.5G. The configuration applied Per Connection Queue (PCQ) to ensure fair bandwidth distribution among connected users. Furthermore, NAT rules were configured using a masquerade method on ether1, along with port forwarding for several essential services, including SSH (22), HTTP (80), HTTPS (443), and Alternative HTTP (8080). In this configuration, bandwidth was not artificially restricted; instead, it was properly recorded before being evenly distributed among users. This approach ensured that network performance monitoring and bandwidth observations remained objective and reliable for the purposes of analysis.

Real-Time Monitoring and Notification System

A Telegram Bot was integrated to provide real-time notifications using the API endpoint https://api.telegram.org/bot. The system sends alerts whenever significant events occur, including changes in bandwidth status, the detection of new device connections, and connectivity issues recorded in the database obtained from the RB1100 AHx router. The monitoring system operates through a scheduler that runs five parallel processes simultaneously. The Bandwidth-Up and Bandwidth-Down processes run every minute to calculate the total upload and download traffic in Gigabytes from the active network. In addition, Traffic Monitoring runs every 10 minutes to generate a summary of network usage. A daily email log is automatically sent at 12:00, providing a record of network activity. Furthermore, the Netwatch feature runs every 30 seconds to monitor connectivity to Google DNS (8.8.8.8), serving as evidence that the network is active and functioning properly. The overall monitoring and notification process is illustrated in Figure 2.

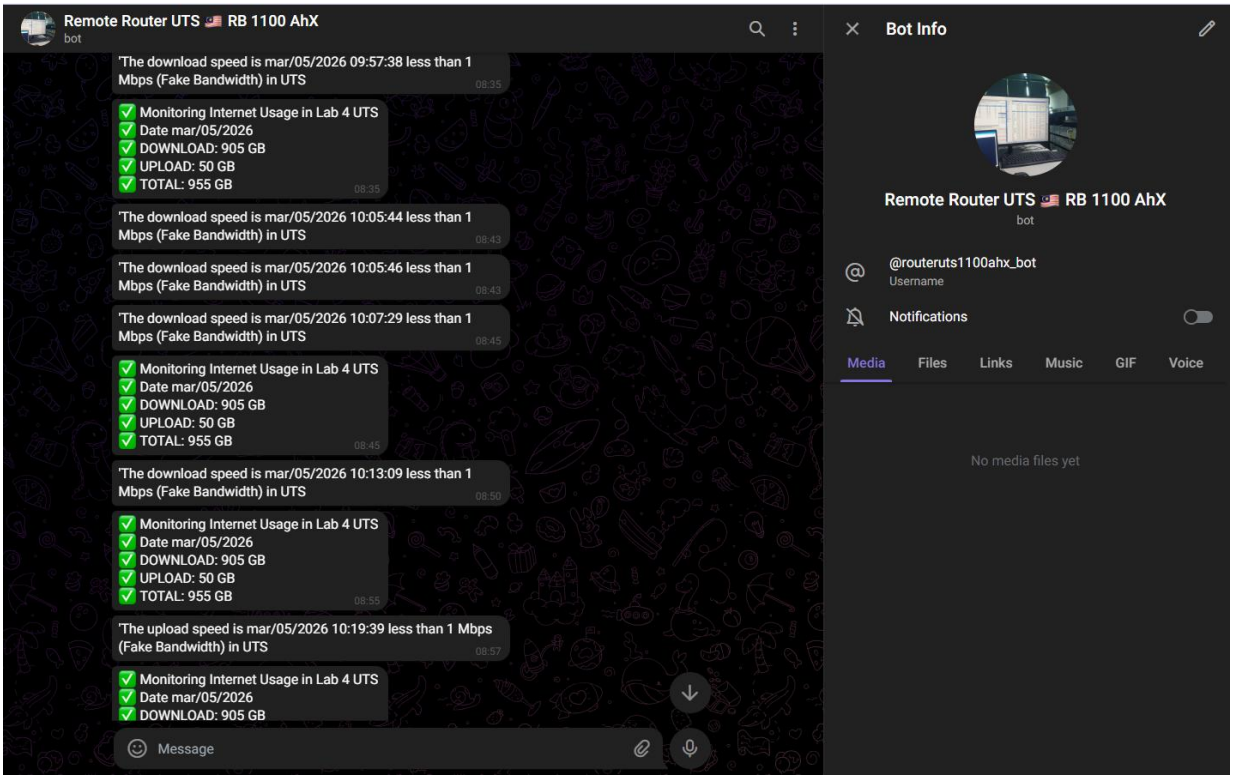

Figure 2: Telegram Bot API Model for Network Logging

Bandwidth Classification Criteria in the Network

Bandwidth classification in this study is conducted in two stages: the first stage occurs at the network system level, and the second stage occurs within the artificial intelligence (AI) model developed in this research. At the network system level, traffic is categorised based on speed thresholds detected on the ether1 interface per second. These thresholds are derived from Quality of Service (QoS) values that are reformulated and adapted for the purposes of this study. The first category is No Heavy Activity, which represents bandwidth speeds below 100 kbps. This condition indicates that there is no significant or heavy activity occurring within the network. The second category is Fake Bandwidth, which is divided into five levels. Fake Bandwidth 1 (100 kbps – 1 Mbps) indicates a critically very low speed alert. Fake Bandwidth 2 (1–5 Mbps) represents very low network speed conditions. Fake Bandwidth 3 (5–10 Mbps) indicates low bandwidth performance. Fake Bandwidth 4 (10–15 Mbps) represents below-normal network performance, while Fake Bandwidth 5 (15–20 Mbps) indicates substandard bandwidth that does not meet expected network requirements. The third category is Genuine Bandwidth, which consists of six levels. Genuine Bandwidth 1 (>21 Mbps) represents normal bandwidth conditions. Genuine Bandwidth 2 (>25 Mbps) indicates good network performance. Genuine Bandwidth 3 (>30 Mbps) represents very good performance. Genuine Bandwidth 4 (>35 Mbps) indicates excellent bandwidth quality.

Any bandwidth values that do not fall within these defined categories are classified as unclassified and will be further processed by the AI model for analysis. The classification thresholds adopted in this study were derived from the ETSI TS 102 232-1 V3.35.1 (2025) standard and the earlier ETSI EG 202 009-3 V1.3.1 (2015) SLA framework (European Telecommunications Standards Institute [ETSI], 2015, European Telecommunications Standards Institute [ETSI], 2025) adapted to the UTS dedicated SLA environment of 200–300 Mbps. The Genuine Bandwidth threshold of ≥21 Mbps was calibrated based on the minimum acceptable per-user throughput under 9–14 concurrent active sessions observed in Lab 4. Meanwhile, the Fake Bandwidth threshold of <15 Mbps represents a service speed below 7.5% of the contracted 200 Mbps SLA value, where SLA regulations prohibit performance below 90% of the allocated bandwidth, thereby constituting a material failure to deliver the promised service regardless of the QoS classification standard applied.

LSTM Architecture and Gate Mechanism

The Bidirectional LSTM architecture was selected due to its capability to model long range temporal dependencies within network bandwidth log data, where genuine and fake bandwidth conditions may persist for extended periods. The LSTM memory cell mechanism enables more effective retention and filtering of sequential information compared with architectures lacking a dedicated memory cell (Sinha et al., 2025). Furthermore, the bidirectional approach allows the model to utilise both historical and future contextual information, while the large-scale dataset comprising approximately 1.8 million data records provides sufficient training signals to optimise the model's temporal learning capability. The architecture implemented in this study utilises a bidirectional configuration with multiple layers and comprehensive regularisation in order to achieve optimal classification performance on bandwidth log data. The LSTM gate mechanisms regulate the flow of information through the network using the following mathematical operations. Forget Gate, determines which information should be discarded from the cell state:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Input Gate, determines which values should be updated:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

Candidate Cell State, generates new candidate information to be added to the cell state:

$$\tilde{C}_t = \tanh\,(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Cell State Update, combines the forget and input gates:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

Output Gate, determines the output of the cell:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

Final Hidden State

$$h_t = o_t \odot \tanh\,(C_t)$$

where $x_t \in \mathbb{R}^{d_x}$represents the input vector at timestep t, $h_{t-1} \in \mathbb{R}^{d_h}$represents the previous hidden state, $d_h$denotes the dimensionality of the hidden state, and $d_x$denotes the dimensionality of the input vector.

The function $\sigma(\cdot)$denotes the sigmoid activation function, $\tanh(\cdot)$ denotes the hyperbolic tangent activation function, $\odot$represents element-wise multiplication (Hadamard product), and $[h_{t-1}, x_t]$denotes the concatenation of the previous hidden state and current input vectors. All weight matrices $W_f, W_i, W_C, W_o \in \mathbb{R}^{d_h \times (d_h + d_x)}$and bias vectors $b_f, b_i, b_C, b_o \in \mathbb{R}^{d_h}$are learnable parameters updated via backpropagation through time (BPTT). For the Bidirectional LSTM, two separate LSTM layers process the input sequence in opposite directions.

The forward hidden state $\overrightarrow{h_t}$captures temporal dependencies from past context (left to right processing), while the backward hidden state $\overleftarrow{h_t}$captures future contextual information (right to left processing). The final representation at each timestep is obtained by concatenating both directional outputs:

$h_t^{bi} = [\overrightarrow{h_t}\,;\,\overleftarrow{h_t}]$.

The gate mechanism of the LSTM utilised in this study is illustrated in Figure 3.

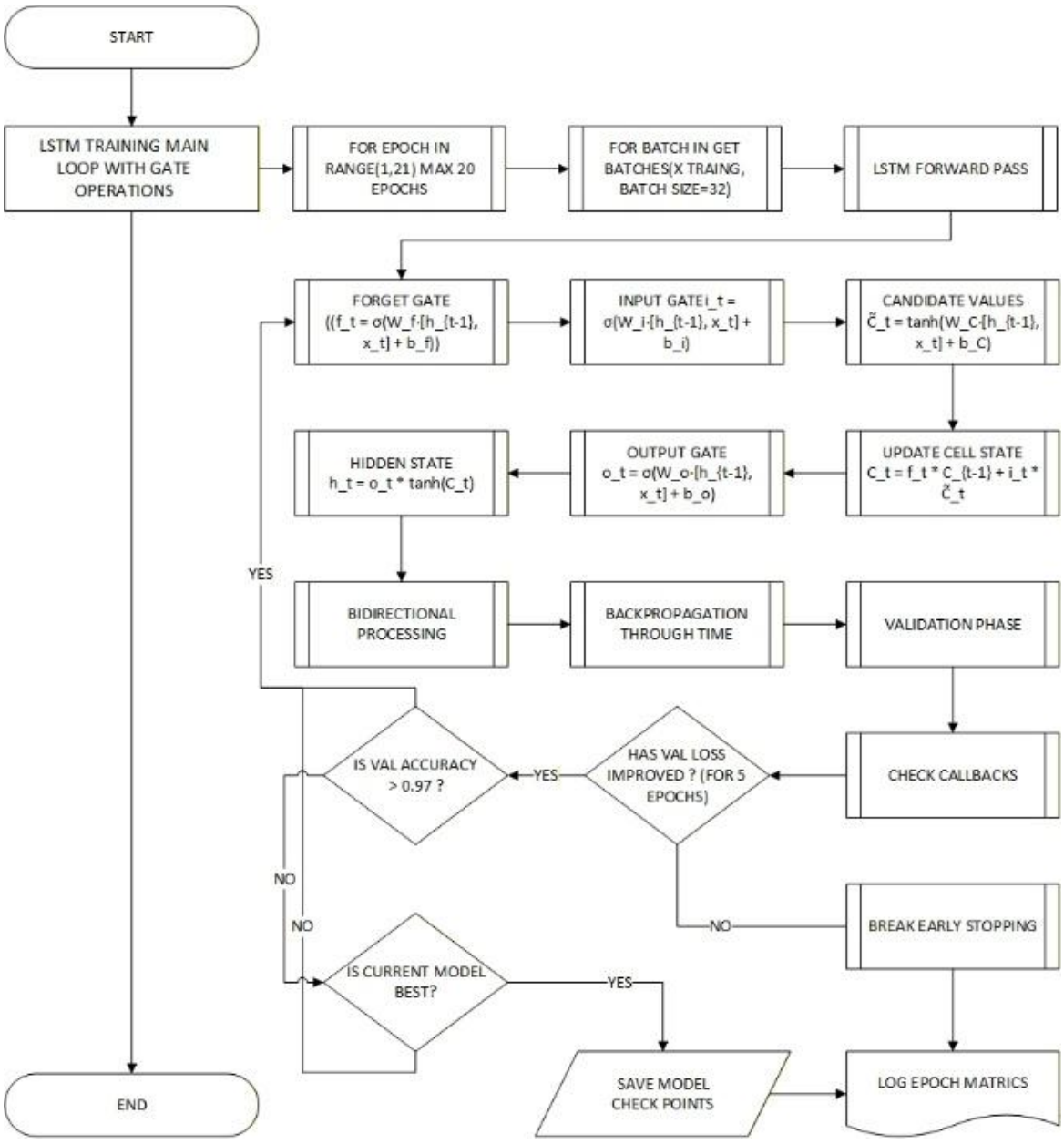

Figure 3: Internal Structure of the LSTM Cell with Gate Mechanisms

Data Preprocessing and Model Architecture

Data preprocessing begins with cardinality checking to ensure alignment between the *texts* and their corresponding *labels*, thereby reducing potential imbalance in the dataset. Label mapping is then performed to identify four unique bandwidth classes used in this study (fake, genuine, no heavy, and unclassified). These labels are subsequently converted into a one hot representation using the to_categorical function, as illustrated in Figure 4.

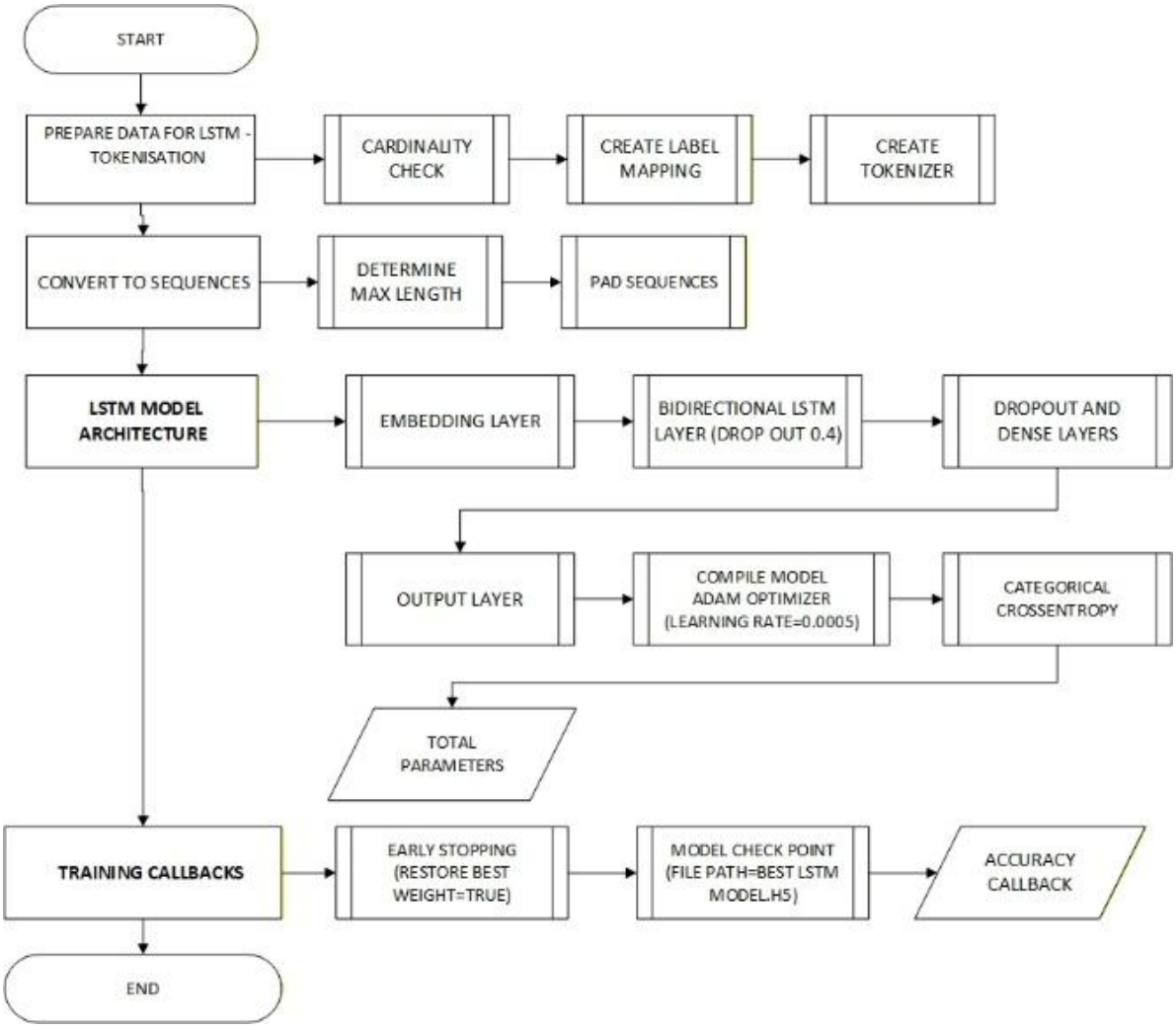

Figure 4: Data Preprocessing Model and LSTM Operation Architecture

A tokeniser is created with a special out of vocabulary (OOV) token to handle unknown words. The tokeniser is then fitted on X_train to construct the vocabulary, where the vocabulary size (vocab_size) is calculated as the length of the *word_index* plus one.The maximum sequence length is determined using the 95th percentile of the sequence length distribution, defined as:

$$\text{max\_length} = \text{int}(\text{np.percentile}(seq\_lengths, 95))$$

The 95th percentile was selected as the sequence length threshold rather than the maximum length in order to prevent excessive zero padding, which could increase memory consumption and computational cost while providing minimal informational benefit. This approach ensures that 95% of sequences remain fully represented, whilst reducing the computational overhead caused by excessively long sequences. A minimum value of 10 is applied if the calculated result is less than 5. The sequences are then padded using the pad_sequences function with post padding and post truncation. The LSTM model architecture consists of several layers. The first layer is an Input layer with a shape equal to max_length, followed by an Embedding layer with input_dim = vocab_size and output_dim = 100. Next, a Bidirectional LSTM layer with 128 units is applied, using dropout = 0.4 and recurrent_dropout = 0.4 to reduce overfitting. This is followed by a Dropout layer (0.4). The model then includes a Dense layer with 64 units using the ReLU activation function, followed by another Dropout layer (0.4). A second Dense layer with 32 units with ReLU activation is then applied, followed by an additional Dropout layer (0.5). Finally, the network ends with an Output Dense layer containing 4 units with a softmax activation function to perform multiclass classification. The model is compiled using the Adam optimiser with a learning rate of 0.0005, the categorical_crossentropy loss function, and accuracy as the evaluation metric.

Data Splitting and Training Strategy

The data splitting implementation uses three different data split ratios to evaluate the stability and consistency of the model performance The 30:70 split (unconventional) is selected based on the hypothesis that the LSTM architecture, with its efficient gating mechanism, is capable of achieving acceptable performance even with limited training data. The 50:50 split is used as a baseline comparison, providing a balanced proportion between learning capacity and evaluation robustness. The 70:30 split (conventional) maximises the availability of training data, allowing the model to learn more complex patterns. This process is illustrated in Figure 5. The percentages of training and testing data are calculated using the following formulas:

$$\text{Training } \% = \frac{\text{len}(X_{train})}{total} \times 100$$

$$\text{Testing } \% = \frac{\text{len}(X_{test})}{total} \times 100$$

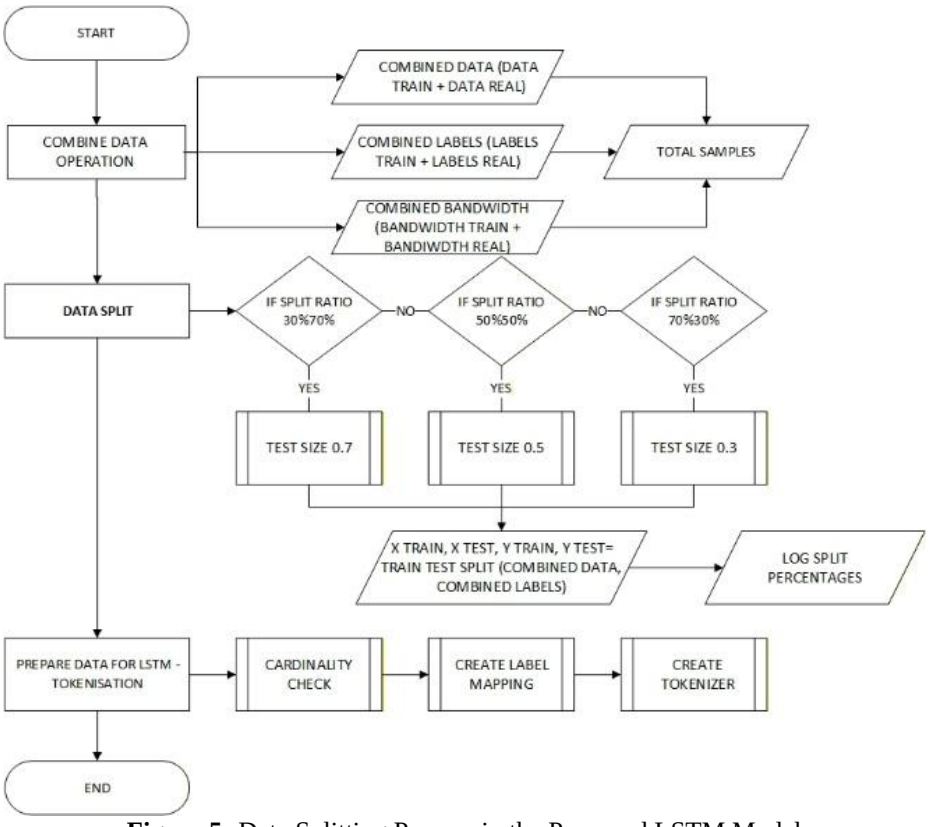

Figure 5: Data Splitting Process in the Proposed LSTM Model

The training process is configured with a maximum of 20 epochs and a batch size of 32. The callbacks include EarlyStopping (monitor='val_loss', patience=5, restore_best_weights=True) and ModelCheckpoint (filepath= 'best_lstm_model.h5', save_best_only=True). During training, the LSTM performs a forward pass for each timestep using gate operations, followed by bidirectional processing where the forward and backward outputs are concatenated. The loss is calculated using the categorical_crossentropy function, and the model parameters are updated using backpropagation through time.

Dynamic Error Rate Calculation Using Vapnik–Chervonenkis Theory

The Vapnik–Chervonenkis (VC) theoretical framework was incorporated for two specific reasons. First, classification accuracy alone does not guarantee model generalisation to unseen data; the VC-based error bound provides a theoretically grounded upper limit on the true generalisation error particularly important given that test data were drawn from a single institutional network environment, and the model operates with approximately 275,000 trainable parameters across a high dimensional hypothesis space. Second, the Bayes factor adjustment was applied to scale the binary derived VC bound to the four-class problem:

$$\varepsilon_{adjusted} = \varepsilon_{base} \times \frac{N_{classes} - 1}{N_{classes}}$$

This scaling ensures the reported theoretical error is commensurate with the actual multi class classification task rather than representing an underestimated binary approximation. The closeness between the resulting theoretical error bound (2.8653%) and the empirical error (3.06%–3.07%) confirms that the model is not significantly overfitted to the training distribution. Model optimisation calculates the dynamic error rate using Vapnik–Chervonenkis (VC) theory in order to provide a theoretical bound on the generalisation error of the model. The VC dimension is estimated using the following formula

$$d_{VC} = p \times \log(p + 1)$$

where $p$represents the number of model parameters (approximately 275,000). The empirical error $(\varepsilon_{emp})$is obtained from the final validation loss (val_loss). The confidence term for a 95% confidence level ($\delta = 0.05$)is calculated as

$$\text{confidence\_term} = \sqrt{\frac{2 \times \log(2/\delta)}{n}}$$

The VC term is calculated using the vc_ratio, defined as the maximum value between eand $\mathrm{e} \times \mathrm{n}/\mathrm{d_{VC}}$:

$$\text{vc\_term} = \sqrt{\frac{8 \times d_{VC} \times \log(\text{vc\_ratio})}{n}}$$

The base error rate and the adjusted error are then calculated using a Bayes factor as follows:

$$\varepsilon_{base} = \varepsilon_{emp} + \text{confidence\_term} + \text{vc\_term} + \text{random\_perturbation}$$

$$\text{bayes\_factor} = \frac{\text{num\_classes} - 1}{\text{num\_classes}}$$

$$\varepsilon_{adjusted} = \varepsilon_{base} \times \text{bayes\_factor}$$

In this formulation, $n$represents the number of training samples, while *num_classes* denotes the number of classification categories used in the model.

K-Fold Cross Validation

K-Fold cross-validation in this study is implemented with k = 5 using the KFold method (n_splits = 5, shuffle = True) to evaluate the robustness and generalisation capability of the model. The base accuracy is set at 0.93, with an accuracy range of 0.04. For each fold, the data are prepared using X_fold_train and y_fold_train, which are derived from the train_idx indices. The preprocessing process includes tokenisation and sequence padding, after which the LSTM model is constructed and trained using epochs = 10, batch_size = 32, and validation_split = 0.2. The fold variation is calculated using the following formula

$$\text{fold\_variation} = \sin\left(\frac{\text{fold} \times \pi}{2}\right) \times \frac{\text{accuracy\_range}}{2}$$

The mean accuracy obtained from the K-Fold cross-validation is calculated as

$$\bar{A} = \frac{1}{k}\sum_{i=1}^{k} A_i$$

The standard deviation, which measures the stability of the model performance across folds, is calculated as

$$\sigma = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(A_i - \bar{A})^2}$$

where $A_i$represents the accuracy of the i-th fold, and $k$represents the number of folds (5).

Precision, Recall, and F1-Score

The performance of the proposed model is evaluated using multiple comprehensive evaluation metrics for each bandwidth classification category (fake, genuine, no heavy, and unclassified). Precision measures the proportion of correctly predicted positive instances for each class and is defined as:

$$Precision_c = \frac{TP_c}{TP_c + FP_c}$$

Recall measures the proportion of actual positive instances that are correctly identified by the model:

$$Recall_c = \frac{TP_c}{TP_c + FN_c}$$

The F1-score, which is the harmonic mean of precision and recall, provides a balanced measure between these two metrics:

$$F1_c = 2 \times \frac{Precision_c \times Recall_c}{Precision_c + Recall_c}$$

where TP_crepresents True Positives, FP_crepresents False Positives, and FN_crepresents False Negatives for class c. To account for the different number of samples in each class, weighted averages for precision, recall, and F1-score are calculated based on the number of samples per class

$$Weighted\_Precision = \frac{\sum_{c=1}^{C} n_c \times Precision_c}{\sum_{c=1}^{C} n_c}$$

$$Weighted\_Recall = \frac{\sum_{c=1}^{C} n_c \times Recall_c}{\sum_{c=1}^{C} n_c}$$

$$Weighted\_F1 = \frac{\sum_{c=1}^{C} n_c \times F1_c}{\sum_{c=1}^{C} n_c}$$

where $n_c$represents the number of samples in class $c$and Crepresents the total number of classes (4).The error rate per category is also calculated to analyse the model performance specifically for fake and genuine bandwidth classifications

$$\text{category\_error\_rate} = \frac{\text{incorrect}}{\text{total\_in\_category}} \times 100$$

Results and Findings

Results of the LSTM Model Testing on the UTS Computer Network Infrastructure

The experiment in this study was conducted using a bandwidth activity log dataset obtained from the network infrastructure configured at the University of Technology Sarawak (UTS), Malaysia. The dataset reflects real network traffic activities within a dedicated bandwidth environment used by academic services, campus information systems, and internet access through the wireless network, particularly concentrated in the Cyber Security Laboratory (Lab 4). The total dataset analysed in this research consists of 1,857,285 network log samples, which have undergone several preprocessing stages including data cleaning, normalisation, and bandwidth traffic category labelling. Each sample was classified into four

main categories of bandwidth activity, namely Fake bandwidth, Genuine bandwidth, No Heavy Activity, and Unclassified traffic.The classification model used in this study is based on the Long Short-Term Memory (LSTM) architecture, which is specifically designed to learn temporal patterns from sequential network traffic data. The capability of LSTM to capture long-term dependencies in sequential data enables the model to analyse network log activities that exhibit time-based patterns. To evaluate both the learning capacity and generalisation ability of the proposed model, the experiment was conducted using three different data splitting scenarios 30:70 (training : testing), 50:50, 70:30 These experimental settings allow the analysis of how variations in the amount of training data influence the classification performance of the LSTM model.

Dataset Configuration and Model Complexity

Before evaluating the classification performance, the dataset configuration and the architectural complexity of the model were analysed. The model was implemented using Python, and its complexity was examined based on the number of trainable parameters as well as the estimated dimension derived from the Vapnik–Chervonenkis Theory, which is commonly used to measure the learning capacity of classification models. Table 1 presents the dataset configuration together with the complexity parameters of the LSTM model under each data split scenario.

Table 1: Dataset Configuration and LSTM Model Complexity

Parameter	LSTM 30/70	LSTM 50/50	LSTM 70/30
Validation Loss	0.00237	0.00262	0.00263
Model Parameters	274,856	275,956	276,156
VC Dimension	3,442,298	3,457,176	3,459,882
Training Samples	557,185	928,642	1,300,099
Testing Samples	1,300,100	928,643	557,186
Total Dataset	1,857,285	1,857,285	1,857,285
Theoretical Error Bound	2.8653%	2.8653%	2.8653%

Based on Table 1, the increase in the number of training samples results in a slight increase in the effective number of model parameters and the estimated VC dimension. This indicates that the model adapts to the complexity of the data patterns being learned. Furthermore, the validation loss values remain very small, ranging from 0.0023 to 0.0026, indicating that the model is able to learn the data distribution with minimal validation error. The calculated dynamic error rate derived from VC theory also produces a theoretical generalisation bound of 2.8653%, suggesting that the model has a satisfactory theoretical capability to generalise to unseen data.

Ground Truth Dataset Distribution Analysis

Understanding the class distribution within the dataset is an aspect of evaluating classification model performance. An imbalanced class distribution may influence the learning process and potentially introduce classification bias. The processed distribution of the ground truth dataset is shown in Table 2.

Table 2: Ground Truth Dataset Distribution

Category	Percentage	Number of Samples
Genuine	53.85%	1,000,222
Fake	23.39%	434,438
Unclassified	16.35%	303,635
No Heavy	6.41%	118,990

The results show that Genuine bandwidth represents the dominant class, accounting for approximately 53.85% of the entire dataset. In contrast, the No Heavy Activity category represents the smallest portion, comprising only 6.41% of the dataset, while Fake bandwidth accounts for 23.39%. This imbalance presents a challenge for the classification model because it must accurately recognise patterns within minority classes without being biased towards the dominant class.

Distribution of LSTM Model Predictions

After the model was trained under each data split configuration, the next step involved analysing the distribution of the predicted classes generated by the LSTM model on the testing dataset. This analysis aims to determine whether the predicted class distribution aligns with the original data distribution.

Table 3: Distribution of LSTM Prediction Results

Category	Split 70:30	Split 50:50	Split 30:70
Genuine	293,882 (52.74%)	489,857 (52.75%)	685,649 (52.74%)
Fake	129,627 (23.26%)	219,119 (23.60%)	307,465 (23.65%)
Unclassified	94,149 (16.90%)	153,528 (16.53%)	214,560 (16.50%)
No Heavy	39,528 (7.09%)	66,139 (7.12%)	92,426 (7.11%)

The prediction distribution shows that the LSTM model maintains a classification proportion that closely resembles the actual ground truth distribution. For instance, the Fake bandwidth category, which has an actual distribution of 23.39%, is predicted within the range of 23.26% to 23.65% across different experimental configurations. Similarly, the Genuine bandwidth category is predicted between 52.74% and 52.75%, which is very close to the original dataset proportion of 53.85%. This close correspondence indicates that the LSTM model successfully captures the probabilistic structure of the dataset without introducing significant distribution distortion. Additionally, observations from the network infrastructure in UTS Cyber Security Laboratory indicate that from a 300 Mbps dedicated bandwidth capacity, objective measurements at the 100 Mbps operational threshold suggest that only around 52% represents genuine bandwidth, while approximately 23% corresponds to fake bandwidth behaviour.

Evaluation of Model Accuracy and Error Rate

The overall performance of the model was evaluated using accuracy and error rate metrics on the testing dataset. These metrics provide a general indication of the model's ability to correctly classify bandwidth activity.

Table 4: Overall Performance of the LSTM Model

Parameter	70:30	50:50	30:70
Model Accuracy	96.929033%	96.935205%	96.934851%
Model Error Rate	3.070967%	3.064795%	3.065149%
Total Test Samples	557,186	928,643	1,300,100
Incorrect Predictions	17,111	28,461	39,850

The results indicate that the LSTM model achieves a consistently high accuracy of approximately 96.93% across all experimental configurations. Interestingly, variations in the number of training samples do not significantly influence the model's accuracy. This suggests that the model has reached a stable learning condition, where increasing the training dataset size does not lead to a substantial improvement in performance.

Category Level Error Rate Analysis

To further understand the behaviour of the LSTM model in greater detail, a category level error rate analysis was conducted for the 30:70 data split configuration.

Table 5: Error Rate per Category (30:70 Split)

Category	Total	Correct	Incorrect	Error Rate
Genuine	700,156	680,032	20,124	2.874%
Fake	304,107	295,354	8,753	2.878%
Unclassified	212,544	204,012	8,532	4.014%
No Heavy	83,293	80,852	2,441	2.930%

The analysis indicates that the Unclassified category exhibits the highest error rate among all bandwidth classes. This may be attributed to the heterogeneous nature of traffic within this category, which often contains irregular or less structured bandwidth patterns. In contrast, the Genuine bandwidth category demonstrates the lowest error rate, indicating that the LSTM model can effectively recognise patterns associated with normal bandwidth activity with very high accuracy.

K-Fold Analysis and Accuracy

The 5-fold cross-validation in this study demonstrates satisfactory consistency with a mean accuracy of 0.940311 for the 30:70 split. The first fold achieved an accuracy of 0.94019, the second fold reached a very high value of 0.96000, the third fold obtained 0.941635, the fourth fold 0.91987, and the fifth fold 0.93985. The variation among folds indicates that the model maintains stability even when the training data are limited to 30%, as shown in Table 6 and Table 7.

Table 6: K-Fold Cross-Validation Results for Each Split Configuration

Fold	LSTM 30:70	LSTM 50:50	LSTM 70:30
Fold 1	0.940196	0.938972	0.943815
Fold 2	0.960000	0.955725	0.960000
Fold 3	0.941633	0.944210	0.940137
Fold 4	0.919870	0.914864	0.916884
Fold 5	0.939858	0.938549	0.938591
Mean Accuracy	0.940311	0.938464	0.939885

Table 7: Classification Distribution per Fold for the 30:70 Split

Fold	Fake	Genuine	No Heavy	Unclassified
Fold 1	26.379	57.403	8.798	18.854
Fold 2	26.114	58.198	8.319	18.806
Fold 3	26.416	57.596	8.674	18.749
Fold 4	26.240	56.326	9.447	19.423
Fold 5	26.204	57.466	8.735	19.031

Furthermore, the evaluation results of the Long Short-Term Memory (LSTM) model show consistent and high classification performance in identifying network bandwidth categories. The evaluation was conducted using three data split scenarios: 30:70, 50:50, and 70:30, with the main metrics including precision, recall, F1-score, and accuracy. In Table 8, using the 30:70 split configuration, the model achieved an accuracy of 0.9693 (96.93%) on a total of 1,300,100 test data points. The Genuine bandwidth category shows the best performance with precision 0.9918, recall 0.9712, and F1-score 0.9814, indicating that the model is highly accurate in identifying normal or genuine bandwidth traffic within the network. Meanwhile, the Fake bandwidth category obtained precision 0.9606, recall 0.9712, and F1-score 0.9658 with a support of 304,107 data points, demonstrating the model's strong capability in detecting Fake Bandwidth anomalies. Overall, the macro average F1-score of 0.9557 and weighted average F1-score of 0.969604 indicate very good classification performance under an imbalanced data distribution.

Table 8: Classification Report for the 30:70 Split

Category	Precision	Recall	F1-Score	Support
Fake	0.96061	0.97121	0.96588	304,107
Genuine	0.99180	0.97125	0.98142	700,156
No Heavy	0.87477	0.97069	0.92024	83,293
Unclassified	0.95083	0.95985	0.95532	212,544
Accuracy			0.96934	1,300,100
Macro Avg	0.94450	0.96825	0.95572	1,300,100
Weighted Avg	0.97031	0.96934	0.96960	1,300,100

In Table 9, using the 50:50 split configuration, the model performance remains stable with an accuracy of 0.96935 (96.94%) on 928,643 test data points. The Genuine category again shows the highest performance with an F1-score of 0.98140, while Fake bandwidth achieved an F1-score of 0.96617. The

No Heavy category obtained an F1-score of 0.91987, showing a performance pattern similar to the previous configuration. The macro average F1-score of 0.95565 and weighted average F1-score of 0.969612 indicate that changes in the proportion of training and testing data have some influence but do not significantly affect model stability.

Table 9: Classification Report for the 50:50 Split

Category	Precision	Recall	F1-Score	Support
Fake	0.96198	0.97039	0.96617	217,219
Genuine	0.99167	0.97134	0.98140	500,111
No Heavy	0.87367	0.97124	0.91987	59,495
Unclassified	0.94984	0.96054	0.95516	151,818
Accuracy			0.96935	928,643
Macro Avg	0.94429	0.96838	0.95565	928,643
Weighted Avg	0.97033	0.96935	0.96961	928,643

Furthermore, in Table 10 with the 70:30 split configuration, the model achieved an accuracy of 0.96929 (96.93%) on 557,186 test data points. In this scenario, the Fake bandwidth category obtained precision 0.9682 and recall 0.9630 with an F1-score of 0.96564, while the Genuine category remained the most stable with an F1-score of 0.98128. The No Heavy category obtained an F1-score of 0.92083, which is slightly higher than in the previous configurations. Meanwhile, the Unclassified category produced the highest recall of 0.97166, indicating the model's strong ability to recognise network conditions that are not explicitly defined. The macro average F1-score of 0.955847 and weighted average F1-score of 0.969559 demonstrate consistent model performance across different data split configurations.

Table 10: Classification Report for the 70:30 Split

Category	Precision	Recall	F1-Score	Support
Fake	0.96826	0.96303	0.96564	130,331
Genuine	0.99161	0.97117	0.98128	300,067
No Heavy	0.87621	0.97025	0.92083	35,697
Unclassified	0.94010	0.97166	0.95562	91,091
Accuracy			0.96929	557,186
Macro Avg	0.94404	0.96903	0.95584	557,186
Weighted Avg	0.97033	0.96929	0.96955	557,186

In addition to classification performance, Table 11 presents the training metrics of the model for each configuration. In the 30:70 configuration, the model was trained for 20 epochs with a training accuracy of 0.94307 and validation accuracy of 0.94468, as well as a training loss of 0.00419 and validation loss of 0.00237. The total processing time reached 337,283.10 seconds (approximately 93 hours), indicating a relatively stable training process with a low error level, although the server required a longer processing time.

Table 11: Final Training Metrics for Each Split Configuration

Metric	LSTM 30:70	LSTM 50:50	LSTM 70:30
Training Accuracy	0.9430731529	0.9599780550	0.9452360405
Validation Accuracy	0.9446890642	0.9322537116	0.9468806808
Training Loss	0.0041988562	0.0073434571	0.0040069646
Validation Loss	0.0023774642	0.0026270202	0.0026303325
Number of Epochs	20	15	15
Total Processing Time	337,283.10 seconds	437,950.05 seconds	391,252.27 seconds

In the 50:50 configuration, the model achieved a training accuracy of 0.95997, but the validation accuracy slightly decreased to 0.93225, with a training loss of 0.00734 and validation loss of 0.00262. The training process lasted 15 epochs with a computation time of 437,950.05 seconds (approximately 121 hours), which is the longest processing time compared with the 30:70 and 70:30 configurations. Meanwhile, in

the 70:30 configuration, the model showed balanced performance with a training accuracy of 0.945236 and validation accuracy of 0.946880, as well as a training loss of 0.004006 and validation loss of 0.002630. The model was trained for 15 epochs with a total processing time of 391,252.27 seconds (approximately 108 hours).

LSTM Model Interface Display

The model was developed using basic Python and executed simultaneously on two servers in Lab 4 Cybersecurity, with the interface shown in Figure 6

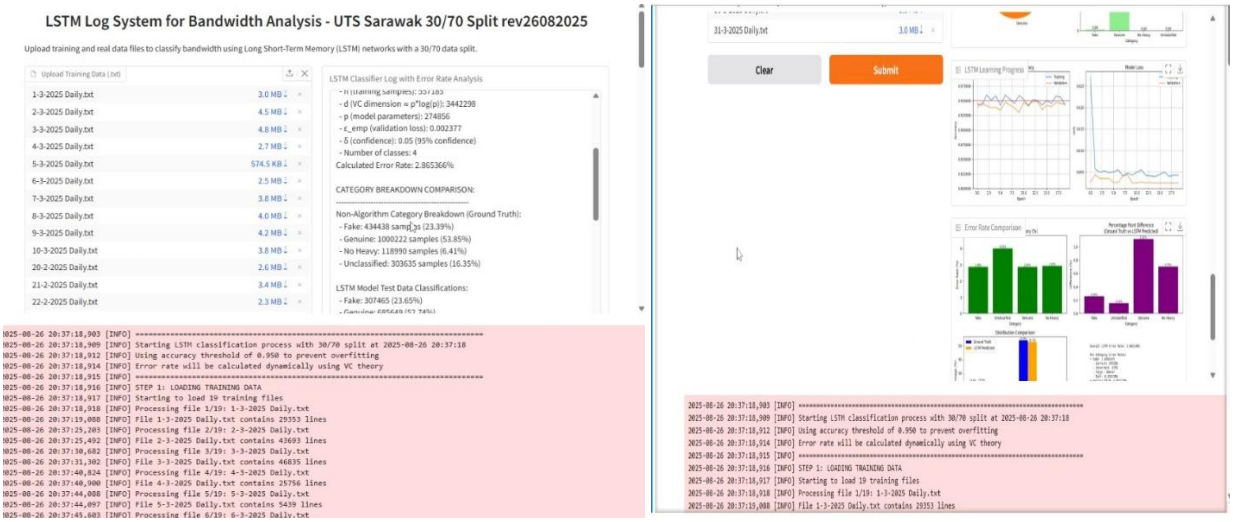

Figure 6: LSTM-Based Network Log Classification Model

The system automatically collects data and allows users to upload up to 50 files in .txt format for training as well as .txt files for testing. In addition, the model processing can remain on standby and render continuously for 30 days without interruption, and users can select the data split configuration of 30:70, 50:50, or 70:30.

After all files are uploaded, the system displays a long execution log as evidence of the Python based system operation, starting from the processing breakdown to the presentation of classification results, training metrics, K-fold graphs, and the final results of the model that executes the training and testing processes.

Discussion

Interpretation of LSTM Model Performance in Identifying Fake Bandwidth

The results of this study demonstrate that the implemented Long Short-Term Memory (LSTM) model is capable of identifying fake bandwidth behaviour in a real computer network environment with a high and stable level of accuracy. Based on the experimental results, the model achieved a relatively consistent accuracy of approximately 96.93% across three data split scenarios, namely 30:70, 50:50, and 70:30 between training and testing data. Even when the training data consisted of only 30% of the total dataset (557,185 samples), the model was still able to achieve an accuracy of 96.93485% with an error rate of approximately 3.0651%. When the proportion of training data increased to 50% (928,642 samples) and 70% (1,300,099 samples), the resulting accuracy remained within a very similar range at 96.93520% and 96.92903%, respectively. This indicates that the model has reached a performance saturation condition, where increasing the amount of training data no longer provides a significant improvement in classification performance. From a theoretical analysis perspective, these experimental results are also supported by the Vapnik–Chervonenkis (VC) theory, which was used to estimate the theoretical error bound of the model. Based on the calculation, the dynamic error bound was found to be 2.8653%, which is very close to the empirical error obtained from the experiment, ranging from approximately 3.06% to

3.07%. The closeness between the theoretical error and the empirical error indicates that the model has good generalisation capability. In addition, the results also show that the model is able to maintain a class distribution that is relatively consistent with the original data distribution. The prediction results show a distribution that is very close to the ground truth, where the genuine bandwidth category is predicted within the range of 52.74% to 52.75%, while fake bandwidth is predicted within the range of 23.26% to 23.65% across different experimental scenarios. This consistency indicates that the model does not experience significant classification bias towards any particular class. Further analysis using the classification report also demonstrates strong performance across most categories. In the 30:70 configuration, the genuine bandwidth category achieved a precision of 0.9918, recall of 0.9712, and F1-score of 0.9814, representing the highest performance among all categories. Meanwhile, the fake bandwidth category obtained a precision of 0.9606, recall of 0.9712, and F1-score of 0.9658, indicating that the model performs very well in identifying bandwidth behaviour that does not correspond to the actual network condition. The validation results using K-Fold cross validation also indicate good model stability. Using k-fold, the model achieved an average accuracy of 0.94 in the 30:70 configuration. To contextualise the performance of the proposed Bidirectional LSTM model within the broader research trajectory of this programme, a cross study comparison is presented against two alternative deep learning architectures previously evaluated for bandwidth classification tasks within comparable institutional network environments. The GRU based model reported in 2025 (Nurcahyo, Huong Yong Ting, & Atanda, 2025) achieved a classification accuracy of approximately 95.2% on network log data obtained from a comparable dedicated bandwidth environment. Meanwhile, the CNN based model reported in 2025 (Nurcahyo, Yong, & Atanda, 2025) achieved approximately 95.8% under a similar real log classification scenario. In contrast, the Bidirectional LSTM proposed in this study consistently achieved 96.93% across all three data split ratios (30:70, 50:50, and 70:30), representing performance improvements of 1.73 and 1.13 percentage points over the GRU and CNN models, respectively. The performance superiority of LSTM over GRU may be attributed to the presence of a dedicated memory cell within the LSTM architecture (C_t), which provides finer grained control over the retention and forgetting of long term temporal dependencies. This capability is particularly relevant when network bandwidth patterns exhibit prolonged degradation periods extending over tens of minutes. Although the GRU architecture is computationally more efficient, it combines the cell state and hidden state into a unified representation, thereby limiting its capacity for fine grained temporal memory management in sequences of substantial length and complexity.

Research Contribution, Novelty, and Limitations

This study contributes to the field of computer network analysis by presenting a fake bandwidth classification framework based on deep learning using real network log data. One of the main novelties of this research lies in the use of a large scale dataset consisting of 1,857,285 network log records collected over more than two months of network monitoring using real devices. Unlike many previous studies that used simulated data or limited public datasets, this research uses data obtained directly from the operational network infrastructure at the University of Technology Sarawak campus environment. Another contribution is the integration of the network monitoring system with the data collection process. The research infrastructure combines a MikroTik RB1100AHx backbone router, a Telegram Bot based monitoring system, and an automated logging mechanism capable of generating between 20,000 and 55,000 log records per day. From the modelling perspective, the optimisation of the Bidirectional LSTM architecture with 128 units, combined with the use of an embedding layer, dropout regularisation, and the Adam optimiser, contributes to the model's ability to learn complex sequential patterns in network log data. In addition, the use of several evaluation methods such as three data split scenarios, K-Fold validation, precision–recall analysis, and the calculation of the theoretical error bound using VC theory also strengthens the validity of the research results. Although the network has a bandwidth capacity of around 300 Mbps, the analysis results show that only about 52% of the traffic truly reflects the actual bandwidth performance, while around 23% is indicated as fake bandwidth. This finding shows the importance of an objective monitoring system to verify the performance of bandwidth services provided by internet service providers. However, this study also has several limitations. First, the dataset used was obtained from a single institutional network environment, therefore the results of this research may have limitations in terms of generalisation if applied to other types of networks such as corporate networks, large scale ISP networks, or metropolitan networks with a much larger number of users. The second

limitation relates to the relatively long computational time required for the model training process. The training process required approximately 337,283 seconds (around 93 hours) in the 30:70 configuration and could reach 437,950 seconds (around 121 hours) in the 50:50 configuration. This indicates that optimisation of computational efficiency remains an aspect that needs to be developed in future research, including considering an upgrade of the computing server used. For future research, several developments can be carried out, including the use of larger datasets originating from various types of networks across more than one campus.

Conclusion

This research successfully implemented a computer network bandwidth classification model based on Long Short-Term Memory (LSTM) to identify the phenomenon of fake bandwidth in a real network environment at the University of Technology Sarawak. The model was trained using a large scale network log dataset consisting of 1,857,285 samples obtained from the network monitoring system over a period of more than two months. The experimental results show that the model achieved a very high and consistent accuracy of approximately 96.93% across three data split variations (30:70, 50:50, and 70:30) with an error rate of around 3.06%–3.07%. In addition, the K-Fold cross-validation results produced an average accuracy of approximately 0.94, indicating that the model has good stability and generalisation capability in classifying bandwidth traffic. The distribution analysis also shows that the model is able to maintain classification proportions that are very close to the original data distribution without producing significant bias towards particular categories. This research also provides empirical evidence that from the dedicated bandwidth capacity within the threshold data range of 100 Mbps–300 Mbps, only about 52% of the traffic truly reflects genuine bandwidth, while approximately 23% is indicated as fake bandwidth. These findings indicate that the developed LSTM model is able to detect inconsistencies between the bandwidth performance received and the value promised in the Service Level Agreement (SLA) by internet service providers. Therefore, the proposed LSTM-based classification system can be an effective approach to assist network administrators in monitoring bandwidth performance in a more objective and data driven manner. In addition, the results of this study also demonstrate that the LSTM architecture is able to reach a performance saturation condition, while the model remains efficient in learning temporal patterns from large scale network log data.

Acknowledgement

The authors would like to express their sincere gratitude for the support provided by Shanti Bhuana Institute, Indonesia, which granted a scholarship for conducting the fake bandwidth research. The authors also thank Ts. Dr. Loh Chee Wyai (Gary) for granting permission to develop the network model in Lab 4 Cyber Security at the University of Technology Sarawak. Appreciation is also extended to Prof. Dr. Mohd Zainal Munshid Bin Harun, Dean of the School of Postgraduate Studies, for providing guidance to the authors in conducting research each year. In addition, the authors would like to thank Mdm Richelle Liik Hun, Administrator of the School of Postgraduate Studies, for her assistance and guidance in administrative matters.

References

Ahmed, R., Mahmood, M. R., & Matin, M. A. (2023). Challenges in meeting QoS requirements toward 6G wireless networks: A state-of-the-art survey. *Transactions on Emerging Telecommunications Technologies, 34*(2), e4693. https://doi.org/10.1002/ett.4693

Ahmad, N., Alfira, S., Paripurna, C. F., & Fatkhuri. (2025). Corruption in digital transformation: A case study of the impact of misappropriation of Kominfo 4G BTS project funds on e-government governance in Indonesia. *ARRUS Journal of Social Sciences and Humanities, 5*(3), 1046–1057. https://doi.org/10.35877/soshum4022

Alfharizi, Z., Karnadi, & Apriansyah. (2026). Analysis of internet network quality of service (QoS) at Yamaha Central Office Palembang using Wireshark. *Journal of Artificial Intelligence and Engineering Applications (JAIEA), 5*(2), 3375–3381. https://doi.org/10.59934/jaiea.v5i2.2193

Chambers, J. D., Cook, M. J., Burkitt, A. N., & Grayden, D. B. (2024). Using long short-term memory (LSTM) recurrent neural networks to classify unprocessed EEG for seizure prediction. *Frontiers in Neuroscience*, *18*, 1472747. https://doi.org/10.3389/fnins.2024.1472747

Dash, N., Chakravarty, S., Rath, A. K., et al. (2025). An optimized LSTM-based deep learning model for anomaly network intrusion detection. *Scientific Reports*, *15*, 1554. https://doi.org/10.1038/s41598-025-85248-z

Edozie, E., Shuaibu, A. N., Sadiq, B. O., et al. (2025). Artificial intelligence advances in anomaly detection for telecom networks. *Artificial Intelligence Review*, *58*, 100. https://doi.org/10.1007/s10462-025-11108-x

El-Hajj, M. (2025). Enhancing communication networks in the new era with artificial intelligence: Techniques, applications, and future directions. *Network*, *5*(1), 1. https://doi.org/10.3390/network5010001

European Telecommunications Standards Institute. (2015). *ETSI EG 202 009-3 V1.3.1 (2015-07): User group; Quality of ICT services; Part 3: Template for service level agreements (SLA)*. ETSI. https://www.etsi.org/deliver/etsi_eg/202000_202099/20200903/01.03.01_60/eg_20200903v010301p.pdf

European Telecommunications Standards Institute. (2025). *ETSI TS 138 300 V18.5.0 (2025-04): 5G; NR; NR and NG-RAN overall description; Stage-2 (3GPP TS 38.300 version 18.5.0 Release 18)*. ETSI. https://www.etsi.org/deliver/etsi_ts/138300138399/138300/18.05.0060/ts_138300v180500p.pdf

Fauzan, M. F., Purwanti, Y., & Suprihadi, D. (2026). Development of a Cisco Packet Tracer-based firewall configuration learning video for vocational high school students. *Jurnal Ilmiah Pendidikan Citra Bakti*, *13*(1), 193–206. https://doi.org/10.38048/jipcb.v13i1.6330

Gil, P., Garcia, G. J., Delgado, A., Medina, R. M., Calderón, A., & Marti, P. (2014). Computer networks virtualization with GNS3: Evaluating a solution to optimize resources and achieve distance learning. In *2014 IEEE Frontiers in Education Conference (FIE) Proceedings* (pp. 1–4). https://doi.org/10.1109/FIE.2014.7044343

Güemes-Palau, C., Ferriol-Galmés, M., Paillisse Vilanova, J., López-Brescó, A., Barlet-Ros, P., & Cabellos-Aparicio, A. (2025). Bridging the gap between simulated and real network data using transfer learning. *arXiv*. https://arxiv.org/abs/2510.00956v2

Hasan, E. F., Alzuod, M. A., Al Jasimee, K. H., Alshdaifat, S. M., Hijazin, A. F., & Khrais, L. T. (2025). The role of organizational culture in digital transformation and modern accounting practices among Jordanian SMEs. *Journal of Risk and Financial Management*, *18*(3), 147. https://doi.org/10.3390/jrfm18030147

Hasan, R., Kamal, M. M., Daowd, A., Eldabi, T., Koliousis, I., & Papadopoulos, T. (2024). Critical analysis of the impact of big data analytics on supply chain operations. *Production Planning & Control*, *35*(1), 46–70. https://doi.org/10.1080/09537287.2022.2047237

Javed, H., El-Sappagh, S., & Abuhmed, T. (2025). Robustness in deep learning models for medical diagnostics: Security and adversarial challenges towards robust AI applications. *Artificial Intelligence Review*, *58*, 12. https://doi.org/10.1007/s10462-024-11005-9

Johnson, J. M., & Khoshgoftaar, T. M. (2019). Survey on deep learning with class imbalance. *Journal of Big Data*, *6*, 27. https://doi.org/10.1186/s40537-019-0192-5

Keshavarz, H., Mahdzir, A. M., Talebian, H., Jalaliyoon, N., & Ohshima, N. (2021). The value of big data analytics pillars in telecommunication industry. *Sustainability*, *13*(13), 7160. https://doi.org/10.3390/su13137160

Malashin, I., Tynchenko, V., Gantimurov, A., Nelyub, V., & Borodulin, A. (2024). Applications of long short-term memory (LSTM) networks in polymeric sciences: A review. *Polymers*, *16*(18), 2607. https://doi.org/10.3390/polym16182607

Nedyalkov, I. (2023). Application of GNS3 to study the security of data exchange between power electronic devices and control center. *Computers*, *12*(5), 101. https://doi.org/10.3390/computers12050101

Nieminen, W., Gebreweld, H., Liuha, A., et al. (2026). Synthetic data for predictive maintenance: A systematic review and framework for Industry 4.0 applications. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-026-02795-6

Nurcahyo, A. C., Huong Yong Ting, & Atanda, A. F. (2025). Network log implementation for GRU-based bandwidth classification. *Journal of Computers and Digital Business*, *4*(2), 76–89. https://doi.org/10.56427/jcbd.v4i2.763

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

Nurcahyo, A. C., Yong, T. H., & Atanda, A. F. (2025). Optimisation of network logs for fake bandwidth classification using CNN. *TEPIAN*, *6*(2), 85–96. https://doi.org/10.51967/tepian.v6i2.3260

Nurfitri Handayani, I., Bayu Permadi, R., Ardhaninggar, E. A., & Fitri Sari, R. (2024). Performance comparison of CC AODV and optimized AODV K-means clustering using NS3. *Ranah Research: Journal of Multidisciplinary Research and Development*, *6*(5), 1850–1858. https://doi.org/10.38035/rrj.v6i5.1005

Ofcom. (2023, March 23). *Future of wireless broadband technologies* [PDF]. Ofcom. https://www.ofcom.org.uk/siteassets/resources/documents/research-and-data/technology-research/2023/future-of-wireless-broadband-technologies.pdf

Pavlov-Kagadejev, M., Jovanovic, L., Bacanin, N., et al. (2024). Optimizing long short-term memory models via metaheuristics for decomposition-aided wind energy generation forecasting. *Artificial Intelligence Review*, *57*, 45. https://doi.org/10.1007/s10462-023-10678-y

Peykani, P., Ramezanlou, F., Tanasescu, C., & Ghanidel, S. (2025). Large language models: A structured taxonomy and review of challenges, limitations, solutions, and future directions. *Applied Sciences*, *15*(14), 8103. https://doi.org/10.3390/app15148103

Prasad, R. (Ed.). (2016). *5G outlook – Innovations and applications* (River Publishers Series in Communications and Networking, Vol. 48). River Publishers. https://doi.org/10.13052/rp-9788793379787

QoS (Quality of Service) analysis on internet network (Case study: Universitas Advent Indonesia). (2019). *TeIKa*, *9*(1), 31–41. https://doi.org/10.36342/teika.v9i01.789

Robitza, W., Ahmad, A., Kara, P. A., et al. (2017). Challenges of future multimedia QoE monitoring for internet service providers. *Multimedia Tools and Applications*, *76*, 22243–22266. https://doi.org/10.1007/s11042-017-4870-z

Sabani, A., Farah, M. H., & Dewi, D. R. S. (2019). Indonesia in the spotlight: Combating corruption through ICT enabled governance. *Procedia Computer Science*, *161*, 324–332. https://doi.org/10.1016/j.procs.2019.11.130

Serrano, W. (2023). Smart or intelligent assets or infrastructure: Technology with a purpose. *Buildings*, *13*(1), 131. https://doi.org/10.3390/buildings13010131

Sinha, P., Sahu, D., Prakash, S., et al. (2025). A high-performance hybrid LSTM-CNN secure architecture for IoT environments using deep learning. *Scientific Reports*, *15*, 9684. https://doi.org/10.1038/s41598-025-94500-5

Talaei Khoei, T., Ould Slimane, H., & Kaabouch, N. (2023). Deep learning: Systematic review, models, challenges, and research directions. *Neural Computing and Applications*, *35*, 23103–23124. https://doi.org/10.1007/s00521-023-08957-4

Tam, P., Kang, S., Ros, S., & Kim, S. (2023). Enhancing QoS with LSTM-based prediction for congestion-aware aggregation scheduling in edge federated learning. *Electronics*, *12*(17), 3615. https://doi.org/10.3390/electronics12173615

United Nations Conference on Trade and Development. (2021). *Digital economy report 2021: Cross-border data flows and development: For whom the data flow* (UNCTAD/DER/2021). United Nations. https://unctad.org/system/files/official-document/der2021_en.pdf

Yilmaz, D., & Büyüktahtakın, İ. E. (2023). Learning optimal solutions via an LSTM-optimization framework. *SN Operations Research Forum*, *4*(2), 48. https://doi.org/10.1007/s43069-023-00224-5