

DDoS attack detection on portmap using a bidirectional recurrent neural network method

Wisnu Adi Putra¹, Deris Stiawan^{1,2}, Qurhanul Rizqie¹, Nurul Afifah^{1,2},
Septiani Kusuma Ningrum^{1,2}, Rahmat Budiarto³

¹Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya, Palembang, Indonesia

²Computer Network, Enterprise and Information Security Research Group, Universitas Sriwijaya, Palembang, Indonesia

³Collage of Computing and Information, Al-Baha University, Alaqiq, Saudi Arabia

Article Info

Article history:

Received Apr 23, 2025

Revised Jun 20, 2026

Accepted Aug 28, 2026

Keywords:

Balanced accuracy

Bidirectional RNN

Correlation-based feature
selection

Distributed denial of service

Matthews correlation
coefficient

Portmap

ABSTRACT

Portmap is a protocol used in Unix and Linux operating systems to map port numbers to their associated program numbers. In a distributed denial of service (DDoS) attack on portmap, attackers will exploit a weakness in this protocol to send many malicious or manipulative server-running portmap requests. Due to the connectionless and unauthenticated nature of UDP, the portmap server will attempt to respond to every request it receives. However, attackers often disguise this so that their IP address is hidden or spoofed. As a result, the server will attempt to reply to requests to a spoofed or invalid IP address, using server resources and causing a denial of service for valid users. A DDoS attack on portmap can be quite damaging. Services associated with portmap may become unresponsive, causing significant downtime for applications or services that rely on the protocol. This study uses the CSE-CIC-IDS 2019 dataset, especially the DDoS attack dataset on portmap. This paper also employs the correlation-based feature selection (CSF) approach to extract significant characteristics from the dataset. Bidirectional RNN is used to detect DDoS assaults. Utilizing 30% testing data and 70% training data, the Bidirectional RNN approach can effectively identify DDoS attacks on portmap. Accuracy is 99.9917%, recall is 99.9917%, specificity is 99.9942%, precision is 99.994%, F1-Score is 99.88%, balanced accuracy (BACC) is 99.92%, and Matthews correlation coefficient (MCC) is 99.98%. By merging the CFS feature selection technique with the Bidirectional RNN architecture, this work mainly contributes to creating a more accurate and efficient DDoS attack detection method. With this method, the detection process can be carried out faster and more precise, reducing the computational burden without sacrificing detection performance.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Deris Stiawan

Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya

Ogan Ilir-30662, Indralaya, Palembang, Indonesia

Email: deris@unsri.ac.id

1. INTRODUCTION

The sustainability of digitalization refers to a long-term transformation process that requires serious attention to security aspects, especially amidst the increasingly complex and less in-depth expansion of cyberattacks. To strengthen system resilience against rising cyber threats, a key challenge in protecting modern digital ecosystems, strategic harmonization of digital transformation with resilient cyber-physical infrastructure becomes an imperative priority. Proactive integration of adaptive security frameworks and

strong network architectures guarantees sustainable operational continuity, countering vulnerabilities aggravated by the spread of complex cyberattacks in today's linked environments. One significant threat is the DDoS attack, which has shown a significant increasing trend. Based on a Kaspersky Lab report, DDoS attacks increased by 47.87% in 2022 [1]. DDoS is a new type of DoS attack that floods server resources such as CPU, memory, and sockets, resulting in unavailability of resources to legitimate users, as well as attacks on the application layer [2]. This phenomenon is increasingly dominant in the IPv4 protocol, with exploitation of Internet service providers (ISPs) and data center operators. The attack traffic pattern that resembles a legitimate client makes its detection a challenge. One method used to detect DDoS attacks on certain protocols such as Portmap is a method based on machine learning [3]. Identification of DDoS attack methodologies can be classified into three main methods, i.e., traditional methods, signature-driven detection, and behavior-based detection, which are two common approaches. Conventional techniques often assess network traffic quantity by comparing it to a predefined baseline. Signature-based detection is done by comparing the observed traffic patterns with signatures in a database, where differences from previously recorded patterns indicate suspicious traffic.

Lastly, anomaly-driven detection systems operate by analyzing typical network activity over a specified timeframe to create a standard behavioral model. Any deviation from this model is flagged as irregular activity, which may signify a potential security breach. This method is very effective in identifying unknown attacks, which do not have a corresponding signature [4]. Literature studies have been conducted to investigate the development of research in the detection of increasingly complex DDoS attacks. Alongside these advancements, several research projects have included machine learning and deep learning to increase the efficacy of attack detection and mitigation. Considering the approach used, existing research works can be categorized into three main aspects, namely the use of hybrid models to improve accuracy, optimization of feature selection to improve efficiency, and the creation of protection mechanisms built on contemporary network topologies such as software-defined networks (SDN).

Hybrid model approaches that combine more than one deep learning technique have been shown to significantly improve the accuracy of DDoS attack detection. Alghazzawi *et al.* [5] suggested a hybrid CNN-BiLSTM model combined with chi-square test for feature selection optimization. Evaluation on the CIC-DDoS2019 dataset showed an accuracy of 94.52%. Amaizu *et al.* [6] created a comparable method with more effective feature selection by combining two deep neural networks (DNN) models. The two DNN models vary primarily in how sequential dropout layers are implemented in each model. The evaluation results showed that this approach successfully achieved an accuracy of 99.66%. Cil *et al.* [7] created a feedforward neural network (FNN) with softmax and sigmoid activation parameters and three hidden layers. The CICDDoS2019 dataset was used to test the model, and it produced a detection accuracy of 99.9% and an attack classification accuracy of 95%.

Shieh *et al.* [8] Additionally, they suggested a hybrid model approach that uses the Gaussian mixture model (GMM) and Bi-LSTM to identify and categorize various DDoS attack types. This method uses GMM to find novel instances that aren't in the training data and Bi-LSTM to differentiate between malicious and genuine traffic. The hybrid Bi-LSTM-GMM architecture achieved precision, recall, and accuracy scores over 94%, demonstrating strong classification performance. The tests use CIC-IDS2017 and CIC-DDoS2019 to assess the suggested model. Rather than improving accuracy, other research focuss on feature selection optimization to increase computational cost efficiency. Fouladi *et al.* [9] suggested an architectural solution that uses continuous wavelet transform (CWT) to identify effective DDoS assaults in an SDN context. The architecture extracts the most relevant features as input for the convolutional neural network (CNN) so that the model is not only more accurate in detecting attacks such as DNS amplification, network time protocol (NTP), and TCP-SYN flooding, but also more computationally cost-effective.

Meanwhile, Wei *et al.* [10] proposed a hybrid deep learning model, autoencoder multi-layer perceptron (AE-MLP) that not only detects but also classifies DDoS attacks in real-time. This model leverages Autoencoder to extract the most relevant features from a large-scale DDoS attack dataset. Feature extraction is utilized for improving detection accuracy while reducing execution time. AE-MLP can help cybersecurity professionals design more optimal mitigation strategies before attacks cause significant impact. The proposed model was tested on the CICDDoS2019 dataset, which includes 5 types of DDoS attacks (LDAP, MSSQL, NetBIOS, SYN, and UDP). The results showed that this model achieves an F1-score of over 98%. Zhou *et al.* [11] suggested a DDoS attack flows identification (SAFE) system that is intended to identify DDoS attack flows in an effective and efficient manner. SAFE may greatly cut down on processing time while increasing attack categorization accuracy. This model's primary benefit is that it preserves user privacy by extracting features without depending on payload examination. Both conventional DDoS assaults and attacks coming from internet of things (IoT) devices may be detected using this technique. Several studies have developed defense systems against DDoS attacks by utilizing SDN, which has emerged as a high-level, centralized network paradigm. SDN enables more adaptive network management, thus

increasing the effectiveness in handling DDoS attacks through a centralized control-based approach and network automation. Assis *et al.* [12] suggested a DDoS attack and intrusion protection system that combines IP flow analysis and a deep learning-based gated recurrent unit (GRU). Based on the evaluation that has been done, the system produces 99.94% accuracy for DDoS detection and 97.09% for intrusion detection.

Varghese and Muniyal [13] proposed a framework called DDoS Detection (D3), a statistical anomaly detection algorithm implemented as a virtual network function (VNF) using open source data plane development kit (DPDK). Experimental results showed that D3 improves the efficiency and effectiveness of the IDS. Almiani *et al.* [14] suggested a Kalman backpropagation neural network-based DDoS intrusion detection model. To show the efficacy of the solution in terms of intrusion detection, the framework was validated using a variety of simulations conducted over the CICDDoS2019 dataset. With an average detection accuracy of 94% and a false alarm rate of 0.0952, the suggested solution also achieved detection rates and precision of 97.49% and 91.22%, respectively. In keeping with the strategy put forward by Yousuf and Mir, the technique is focused on identifying DDoS assaults in IoT contexts. Detecting attack using live capture neural network (DALCNN), a novel way to developing a security system based on adaptive and distributed network architecture, makes use of the OpenDayLight platform, SDN implementation, and the idea of recurrent neural networks [15].

The principal contribution of this paper is a framework that amalgamates BRNN, CFS, and SMOTE, specifically tailored for Portmap-specific DDoS detection. It adeptly captures bidirectional temporal dependencies and mitigates significant class imbalance, challenges often overlooked by traditional unidirectional models like GRU or resource-intensive CNN-based hybrids. Our method utilizes correlation-based feature selection (CFS) to identify the most relevant attributes, thereby reducing computational time while maintaining high sensitivity. This enhancement ensures an optimal balance between processing efficiency and high detection rates, specifically designed for the complex signatures of Portmap traffic. This document is formatted as follows: Section 2 provides a description of the tools and methods used in this study. The research results and remarks related to the performance analysis of the recommended method are included in section 3. Section 4 contains conclusions and recommendations for more research.

2. RESEARCH METHOD

All things considered, the research approach used to construct this paper necessitates a well-defined framework at every turn. The strategy, which is shown in Figure 1, starts with a study of current research in the literature before preparing data using the CIC-IDS-2019 dataset [16]. Data pre-processing via feature extraction and selection comes next, followed by testing, result analysis, and conclusion drafting.

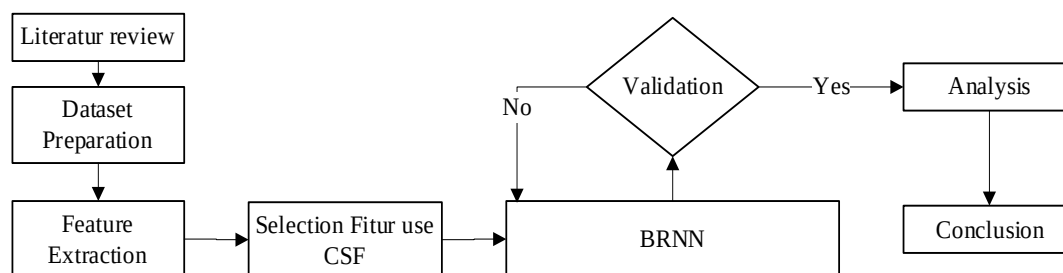


Figure 1. Research method

2.1. Dataset preparation

In this research, the workflow for detecting DDoS attacks targeting Portmap initiates with a data preparation phase, as illustrated in Figure 2. Subsequently, a bidirectional recurrent neural network (BRNN) is integrated to identify DDoS anomalies, leveraging predefined feature sets extracted during preprocessing to train the detection model. Subsequent to model development, a rigorous validation process is conducted to assess the model's predictive consistency and operational robustness. This phase encompasses a diagnostic assessment of validation outcomes, the purpose of which is twofold: first, to evaluate the efficacy of the detection framework, and second, to derive actionable conclusions from the experimental findings.

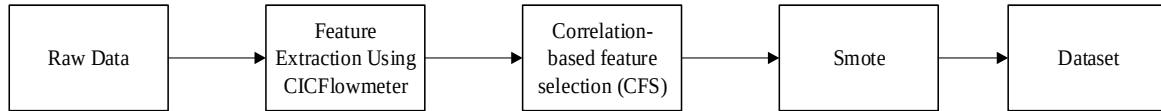


Figure 2. Dataset preparation

2.2. Feature extraction

The preprocessing phase consists of two sequential methodological steps: feature extraction followed by feature selection. During the initial step, the CICFlowmeter V3 tool is employed to systematically extract network traffic characteristics. This process generates a structured dataset in .csv format, where the extracted attributes are organized into distinct feature categories. Subsequently, the feature selection phase is implemented to refine the dataset by identifying and retaining the most discriminative features, thereby optimizing computational efficiency and model performance.

2.3. CSF

In previous research [17], a high correlation value in a feature indicates that the feature is highly relevant. The CSF method [11], [18] with the aim of reducing useless features in detecting DDoS attacks involving portmaps from the dataset. It is anticipated that this feature selection would enhance the model's ability to identify portmap-based DDoS assaults. CSF is a feature selection method where features are selected by looking at the correlation value for each feature. The evaluation function of a subset of features in CFS is obtained using (1).

$$Ms = \frac{k \overline{rcf}}{\sqrt{(k+k)(k-1) \overline{rff}}} \quad (3)$$

where, Ms is the heuristic “merit” of the feature subset S consisting of k features, K is the number of attributes, $\overline{rcf}$ is the mean of the correlation between features and classes ($f \in S$), and $\overline{rff}$ is the average value of the inter-correlation of the features. After data extraction using CICFlowmeter, 83 features were obtained, then the feature selection process was carried out using the CSF method on the dataset, the features in the dataset were reduced from 83 features to 38 best features which can be seen in Table 1.

In Figure 3 thirty-eight features were selected, comprising 186,960 attack data points and 4,734 benign data points. Figure 3(a) illustrates the distribution of this raw data, highlighting a pronounced class imbalance. To mitigate the bias and enhance detection accuracy of the BRNN caused by such an imbalance, we employed the synthetic minority over-sampling technique (SMOTE) to equilibrate the classes. Figure 3(b) illustrates the resultant balanced distribution of the dataset, wherein the minority class has been oversampled to correspond with the majority class.

Table 1. Feature results from the CSF process

#	Attribute	#	Attribute	#	Attribute
1	Source Port	13	Flow IAT Min	26	CWE Flag Count
2	Destination Port	14	Fwd IAT Total	27	Down/Up Ratio
3	Protocol	15	Fwd IAT Min	28	Init_Win_bytes_forward
4	Flow Duration	16	Bwd IAT Total	29	Init_Win_bytes_backward
5	Total Fwd Packets	17	Bwd IAT Min	30	act_data_pkt_fwd
6	Total Length of Fwd Packets	18	Fwd PSH Flags	31	min_seg_size_forward
7	Total Length of Bwd Packets	19	Fwd Header Length	32	Active Mean
8	Fwd Packet Length Min	20	Fwd Packets/s	33	Active Std
9	Fwd Packet Length Std	21	Bwd Packets/s	34	Active Min
10	Bwd Packet Length Min	22	Max Packet Length	35	Idle Mean
11	Bwd Packet Length Std	23	SYN Flag Count	36	Idle Std
12	Flow Bytes/s	24	ACK Flag Count	37	Inbound
		25	URG Flag Count	38	Label

2.4. Bidirectional RNN

The BRNN is an advanced form of unidirectional recurrent neural network (RNN). The RNN usually only focuses on the previous context, whereas BRNN focuses on the context before and after, so it

can use more or more diverse information to make a prediction [19], as illustrated in the architecture diagram in Figure 4.

Formally, multiply the hidden layer activation function by ϕ for each time step represented by t . In this case, the mini-batch input is $X_t \in R_n \times d$, where n is the number of samples and d is the number of inputs in each sample. $H_t \in R_n \times d$ and $\vec{H}_t \in R_n \times d$, and where h is the number of hidden units, are the forward and backward hidden states for this time step in the bidirectional architecture.

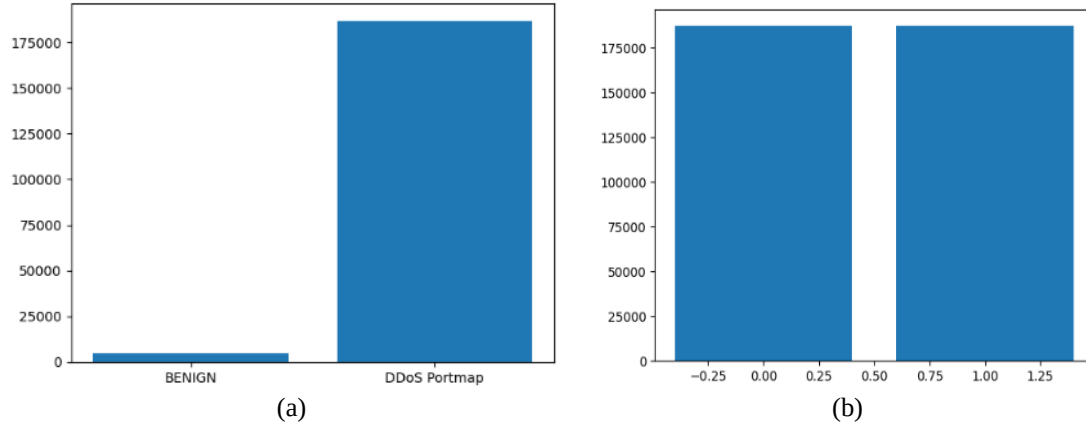


Figure 3. Selection of 38 features for attack detection (a) unbalanced obtained dataset and (b) dataset after SMOTE process

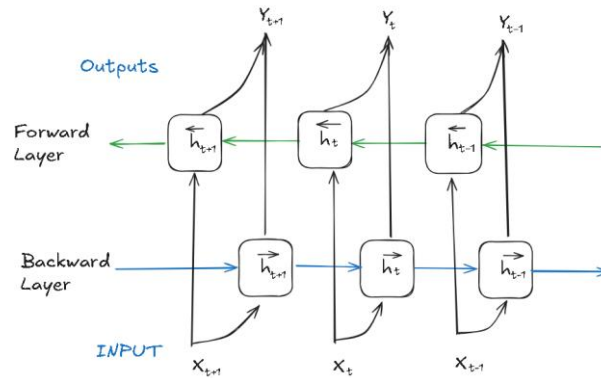


Figure 4. BRNN architecture [20]

The hidden state advances using (2).

$$\vec{H}_t = \phi(X_t W_{xh}^{(f)} + \vec{H}_{t-1} W_{hh}^{(f)} + b_h^{(f)}) \quad (2)$$

The hidden status is reversed using (3).

$$H_t^- = \phi(X_t W_{xh}^{(b)} + H_{t+1}^- W_{hh}^{(b)} + b_h^{(b)}) \quad (3)$$

where $W_{xh}^{(f)} \in R^{d \times h}$, $W_{hh}^{(f)} \in R^{h \times h}$, $W_{xh}^{(b)} \in R^{d \times h}$ and $W_{hh}^{(b)} \in R^{h \times h}$, and $b_h^{(f)} \in R^{1 \times h}$ bias and $b_h^{(b)} \in R^{1 \times h}$ on all parameter models.

The subsequent phase involves consolidating the forward and reverse states $\vec{H}_t$ and H_t^- to derive the hidden state $H_t \in R^{n \times 2h}$ which will then be input into the output layer. In a deep bidirectional recurrent neural network (RNN) composed of multiple hidden layers, information is transmitted as input to the next bidirectional layer. Ultimately, the output layer generates the output $O_t \in R^{n \times q}$ where (q) denotes the output dimension utilizing (4).

$$O_t = H_t W_{hq} + B_q \quad (4)$$

With weight matrix $W_{hq} \in R^{2h \times q}$ and with bias $b_q \in R^{1 \times q}$ is a model parameter of the output layer. Meanwhile, technically, bidirectional can have different numbers of hidden units.

After the preprocessing stage, the methodological workflow moves to model building using the bidirectional recurrent neural network (BRNN) architecture. The dataset is methodically divided into two separate parts—a training set for calibrating the model's parameters via iterative learning and a testing set for evaluating the model's generalization capacity using previously unobserved data—to guarantee strong assessment. This division follows accepted machine learning practices: the training phase maximizes predictive accuracy and the testing phase confirms performance under real-world conditions.

2.5. Model performance measurement

Key measures including accuracy, sensitivity (recall), precision, and F1 score will be used to evaluate the constructed model's efficacy throughout the evaluation phase. A confusion matrix, a tabular tool that compares expected results with actual data labels, is the source of these measures. True Positives (TP, properly identified positives), False Positives (FP, erroneously highlighted positives), True Negatives (TN, accurately recognized negatives), and False Negatives (FN, ignored positives) are the four different classifications into which the matrix divides the findings. These divisions serve as the basis for computing quantitative performance metrics, allowing for a thorough assessment of the predictive accuracy of the model [21], [22].

The level of accuracy is determined by comparing the total correct predictions (including True Positive and True Negative) against all the samples used in the testing. Mathematically, this accuracy level can be expressed through the following (5).

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (5)$$

Sensitivity, also known as recall, measures how well a model finds true positives in a dataset. The number of correctly anticipated positives (true positives) divided by the total number of positive cases which includes both correctly recognized positives (true positives) and undiscovered positives (false negatives) is how this performance measure is computed. This metric may be stated mathematically using (6).

$$Sensitivity = \frac{TP}{TP+FN} \quad (6)$$

Precision serves as a performance indicator to evaluate the reliability of a model's positive predictions in distinguishing target classes within a dataset. This metric is derived by dividing the number of correctly identified positive cases (true positives) by the aggregate count of instances classified as positive, which encompasses both accurate classifications (true positives) and erroneously classified positive instances (false positives). The mathematical representation of precision is provided in (7).

$$Precision = \frac{TP}{TP+FP} \quad (7)$$

The F1-score provides a thorough evaluation of a model's predictive performance by combining precision and recall (sensitivity) into a single metric. This metric assesses how well a model balances the ability to detect all relevant positive occurrences (recall) with the precision of accurately recognizing positive cases (precision). (8) shows how the F1-score formula is obtained from the harmonic mean of these two metrics.

$$F1 \text{ score} = \frac{2*(precision*sensitivity)}{precision+sensitivity} \quad (8)$$

Performance assessment of a detection process is also carried out using the BACC and MCC parameters [23]. BACC is used to measure the level of accuracy of accuracy performance on imbalanced data [24], meanwhile, MCC is used in measuring sensitivity performance on imbalanced data [25]. The equation for the BACC and MCC is shown in (9) and (10), respectively.

$$BACC = \frac{\frac{TP}{P} + \frac{TN}{N}}{2} \quad (9)$$

The equation for MCC can be seen using the following equation (2.12).

$$MCC = \frac{(TP*TN)-(FP*FN)}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}} \quad (10)$$

3. RESULT AND DISCUSSION

At this point, the study findings are acquired, and the model's performance is measured in order to evaluate the findings. The BRNN approach in this work makes use of a number of hyperparameters. The activation functions in the BRNN and output layers, the Loss function, the learning rate function, the Optimizer, and the dropout amount are some examples of these hyperparameters. It is also used to compare the quantity of testing and training data. The goal of this testing is to maximize the model's performance during training before using it for a DDoS attack detection system.

3.1. Scenario 1 based on hidden layer

Table 2 shows the results of the model performance to determine the best value when used in the Hidden layer. In this test, 50% testing data and 50% training data were used, with a batch size of 256, the optimizer function used SGD, and the learning rate used was 0.001 and 50 epochs Based on Table 2, the hidden layer configuration of 64,32,16 demonstrates marginally superior performance, with the highest accuracy of 99.80%. Although precision and recall are consistent across configurations, the 64,32,16 configuration has fewer false positives, indicating a slightly better generalization ability for distinguishing attack.

Table 2. Hidden layer tuning

Hidden layers testing	Precision	Recall	Specificity	Accuracy	TP	TN	FP	FN
128,64,32	99.90%	99.69%	99.89%	99.79%	93569	93002	98	291
64,32,16	99.90%	99.69%	99.90%	99.80%	93575	93002	92	291
32,16,8	99.90%	99.69%	99.89%	99.79%	93569	93002	98	291

3.2. Scenario 2 based on batch size

In the next test, the best value in the previous hidden layer will be used, where it will be tested to determine the best batch size value. Table 3 will show the results of the batch size trial using 50% testing data and 50% training data, SGD optimizer function, hidden layer 64,32,16, 50 epochs, and a learning rate of 0.001. From Table 3, a batch size of 128 yields the highest accuracy at 99.86%, significantly improving recall to 99.81% compared to larger batch sizes. Smaller batch sizes appear to enhance the model's capability to identify true positives and reduce false negatives, likely due to more frequent weight updates and gradient adjustments during training.

Table 3. Batch size tuning

Batch Size	Precision	Recall	Specificity	Accuracy	TP	TN	FP	FN
1024	99.82%	99.69%	99.82%	99.76%	93502	93006	165	287
512	99.89%	99.68%	99.89%	99.78%	93562	92994	105	299
256	99.91%	99.69%	99.91%	99.80%	93589	92998	78	295
128	99.91%	99.81%	99.91%	99.86%	93582	93115	85	178

3.3. Scenario 3 based on drop out

In the next testing, a trial was conducted on the Dropout value. Table 4 shows the results of the Dropout trial using 50% testing data and 50% training data, stochastic gradient descent (SGD) optimizer function, number of Hidden layer 64, 32 and 16; batch sizes: 128 and 50 epochs; and a learning rate of 0.001.

Table 4. Dropout tuning

DropOut Value	Precision	Recall	Specificity	Accuracy	TP	TN	FP	FN
0.1	99.54%	98.83%	99.99%	99.99%	2627	952488	12	31
0.2	99.51%	98.61%	99.99%	99.99%	2626	952482	13	37
0.3	99.51%	99.06%	99.99%	99.99%	2626	952494	13	25
0.4	99.32%	98.28%	99.99%	99.99%	2621	952492	18	19
0.5	99.46%	99.09%	99.99%	99.99%	2625	952495	14	24

According to Table 4, dropout values between 0.1 and 0.3 perform similarly well, with a 0.3 dropout slightly better in recall at 99.06%. However, as dropout increases beyond 0.3, performance slightly decreases, indicating that excessive dropout could hinder learning by overly suppressing neuron activations critical to attack detection.

3.4. Scenario 4 based on activation function

Table 5 shows the results of the trial of the activation function using 50% testing data and 50% training data, the SGD optimizer function, number of Hidden layer: 64, 32 and 16; batch sizes: 128 and 50 epochs; and a learning rate of 0.001. From Table 5, the Sigmoid activation function achieves the highest accuracy and balanced precision-recall trade-off. The Softmax function fails entirely due to misclassification, and the ReLu function significantly reduces precision and recall. The Sigmoid function provides stable and consistent activation suitable for binary classification tasks.

Table 5. Activation function tuning

Activation Function	Precision	Recall	Specificity	Accuracy	TP	TN	FP	FN
Sigmoid	99.54%	98.83%	99.99%	99.99%	2627	952488	12	31
Softmax	0	Nan	99.72%	99.72%	0	952519	2639	0
Softplus	99.58%	98.24%	99.99%	99.99%	2628	952472	11	47
ReLu	95.87%	96.05%	99.99%	99.98%	2530	952415	109	104
Tanh	99.20%	98.60%	99.99%	99.99%	2618	952482	21	37

3.5. Scenario 5 based on learning rate

A trial on the learning rate was also conducted. The learning rate is a speed used to calculate the weight correction value during the data training process. Table 6 shows that a learning rate of 0.001 significantly outperforms lower learning rates. Lower learning rates cause slower convergence and poorer recall, suggesting inadequate updates during training. Conversely, a learning rate of 0.001 achieves rapid convergence and high accuracy, highlighting the importance of appropriate learning rate selection for efficient and effective training.

Table 6. Learning rate tuning

Learning Rate	Precision	Recall	Specificity	Accuracy	TP	TN	FP	FN
0.001	99.54%	98.83%	99.99%	99.99%	2627	952488	12	31
0.0001	97.27%	97.60%	99.99%	99.98%	2567	952456	72	63
0.00001	96.32%	79.41%	99.99%	99.92%	2542	951860	97	659
0.000001	21.10%	2.59%	99.78%	97.59%	557	931596	2082	20923

3.6. Best model

Hyper parameters used include activation function on BRNN layer, type of activation function on output layer, loss function, learning rate function, optimizer function, Dropout size used and comparison of training data and test data. Previously, testing has been done and the best value is also obtained, Table 7 shows the best hyper parameters used in the classification process with the BRNN.

Table 7. The best value of hyper parameters

Parameter	The best result
Epochs	100
Number of BRNN Layer	3
Number of Node BRNN Layer 1	64
Number of Node BRNN Layer 2	32
Number of Node BRNN Layer 3	16
Number of Node Output Layer	1
BRNN Layer Activation Function	tanh
Loss Function	Binary Cross-entropy
Output Layer Activation Function	Sigmoid
Learning Rate	0.001
Optimizer	SGD
Dropout	0.1
Number of Input Layer Node	38

In the model testing experiments, 9 test scenarios were carried out, with several proportional of testing data vs. training data as depicted in Table 8. Having done the testing and obtaining the results, the best model uses a proportion of 70% for training data vs. 30% for training data. Figure 5 shows the accuracy of the model and the loss model, that the value of the loss graph decreases very significantly so that the detection process produces a stable final value. From Figures 5(a) and 5(b) it can be concluded that the graph of the attack detection results shows that the graph is convergent, resulting in good performance during the detection process. The results of the Confusion Matrix in Figure 6 are calculated to obtain the values of accuracy, specificity, recall, precision and F1 Score, and the results can be seen in Table 8.

Table 8. Comparison of test data and training data to test the BRNN model

Validation	Comparison of data amount	Amount of data	
		Training data	Test data
First Validation	10:90	337163	37463
Second Validation	20 :80	299136	74784
Third Validation	30 : 70	261744	112176
Fourth Validation	40 : 60	224352	149568
Fifth Validation	50:50	186960	186960
Sixth Validation	60:40	149568	224352
Seventh Validation	70:30	112176	261744
Eighth Validation	80:20	74784	299136
Ninth Validation	90:10	37392	336528

The accuracy graph in Figure 5 indicates a rapid increase in accuracy values for both training and testing data early in the training process, quickly stabilizing at a high value of approximately 99.917%. The consistency and close proximity of accuracy values between the training and testing data indicate that the BRNN model has successfully converged, demonstrating robust generalization capabilities and no significant signs of overfitting or underfitting in detecting DDoS attacks. The loss graph displays a steep initial decline, followed by a gradual decrease until stabilizing at a minimal value. This indicates the model learns quickly and effectively optimizes network weights, achieving very low prediction errors on both training and testing datasets. In conclusion, both graphs clearly illustrate that the BRNN model demonstrates excellent, stable, and efficient performance in detecting DDoS attacks, characterized by high accuracy and low loss values.

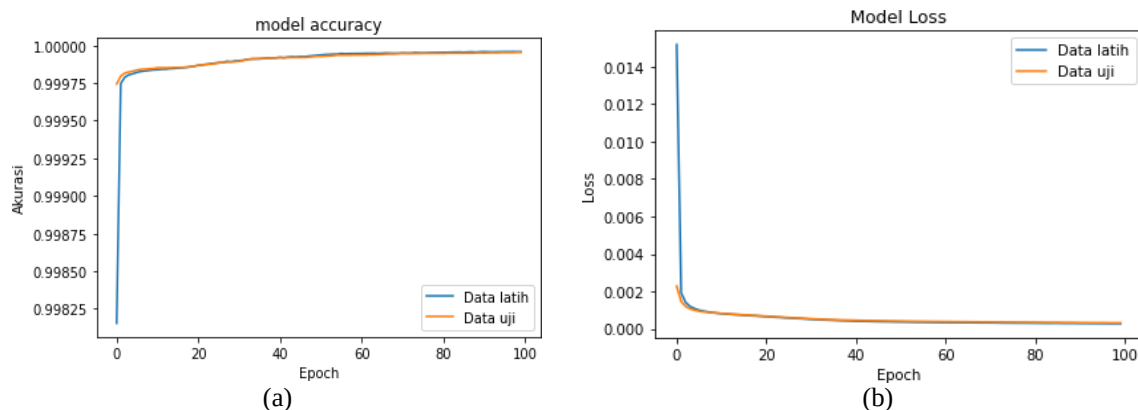


Figure 5. Model accuracy and loss convergence (a) validation of the results of attack detection using the BRNN method obtained an accuracy value of 99.917% and (b) the value of the loss graph experienced a very significant decrease

The minimal incidence of false positives and false negatives in Figure 6, compared to true positives and true negatives, underscores the model's strong predictive efficacy. Table 9 encapsulates the performance metrics derived from a 70:30 training-to-testing data division. The BRNN model effectively differentiates between normal traffic and DDoS attacks, exhibiting exceptional precision and recall in its detection capabilities.

In addition to using parameters from the accuracy, recall, specificity, precision, and F1-Score values, this study also used the BACC and MCC validation approaches as shown in Figure 7. ROC curve and

precision–recall curve demonstrating the model’s classification performance as shown in Figures 7(a) and 7(b). Table 10 shows the performance of BACC value get 99.92% indicates an excellent balance between sensitivity (recall) and specificity, highlighting the model's capability to accurately distinguish between attack and non-attack instances. The MCC value of 99.98% underscores the high reliability and robustness of the model's predictions, further confirming its effectiveness in DDoS attack detection.

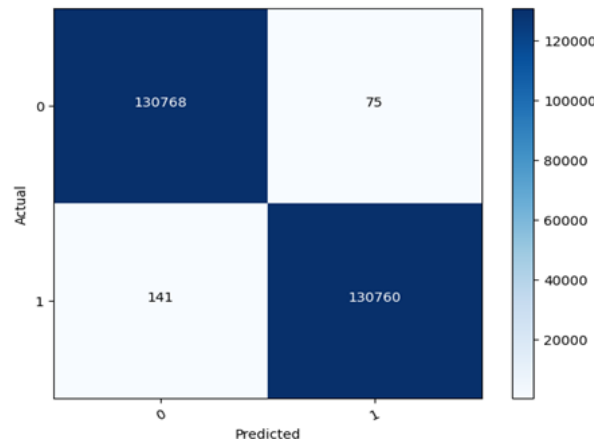


Figure 6. Confusion matrix of training data and test data 70:30

Table 9. Validation results with training data and test data 70:30

Data ratio	Test				
	Accuracy	Recall	Specificity	Precision	F1-Score
70:30	99.99174%	99.99898%	99.99426%	99.99426%	99.8888%

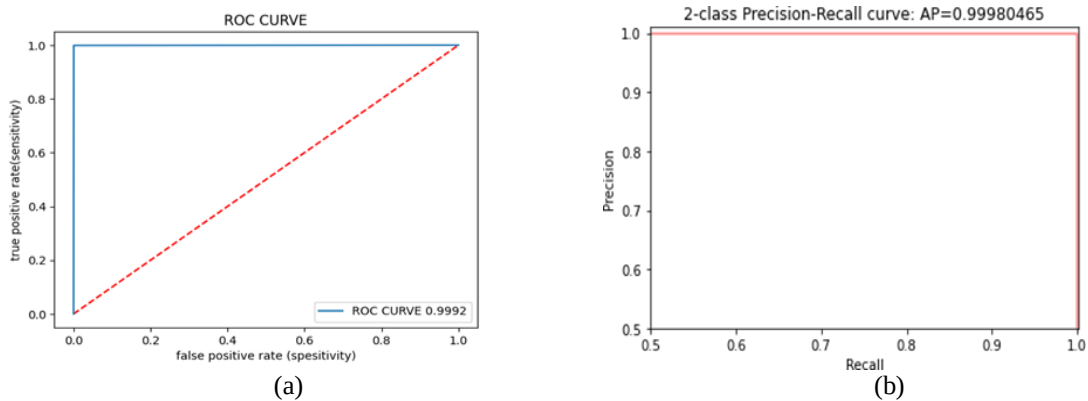


Figure 7. Model validation using BACC and MCC (a) ROC curve graph and (b) value of precision-recall graph

Table 10. BACC validation results and average precision-recall

Validation	Result
BACC	99.92%
MCC	99.98 %

4. CONCLUSION

In conclusion, the results obtained in this research demonstrate outstanding performance of the BRNN model for detecting DDoS attacks on portmap. The use of the CFS method significantly improved the detection process by reducing unnecessary features from 83 to 38, thereby enhancing computational efficiency without sacrificing the accuracy of the detection. The remarkable accuracy 99.9917%, precision

99.994%, recall of 99.9917%, specificity 99.9942%, and an F1-score 99.88% obtained from a 70%-30% data split validates the robustness and reliability of the BRNN architecture. These results can be attributed to several key factors: i) Firstly, the inherent bidirectional structure of BRNN effectively captures complex temporal dependencies in network traffic data, allowing it to accurately identify subtle anomalies that characterize DDoS attacks. Unlike traditional unidirectional models, BRNN can leverage both past and future context, which is crucial for accurately classifying network behavior in real-time scenarios. ii) Secondly, the strategic integration of SMOTE helped mitigate the imbalance issue in the dataset, thereby preventing bias towards the majority class (attack data). By balancing the dataset, the model could generalize better and avoid skewed detection outcomes. iii) Thirdly, the careful tuning of hyperparameters such as the number of hidden layers and nodes, activation functions, dropout rate, optimizer selection, and learning rate significantly contributed to the model's optimal performance. Specifically, the selected combination of tanh activation functions in hidden layers and sigmoid activation in the output layer provided a robust mechanism to handle non-linear characteristics in the dataset effectively.

Moreover, validation metrics BACC and MCC further reinforce the model's reliability, especially in handling imbalanced datasets, with impressive values of 99.92% and 99.98%, respectively. These metrics indicate that the model not only achieves high accuracy but also maintains stability across different classes, effectively distinguishing between legitimate traffic and malicious requests. In summary, the superior results achieved in this study reflect the effectiveness of combining a sophisticated feature selection method, an advanced deep learning architecture, balanced dataset techniques, and meticulous hyperparameter tuning. The proposed method provides a reliable and efficient solution for securing portmap services against evolving DDoS threats.

ACKNOWLEDGEMENTS

The authors would like to thank the Computer Network, Enterprise and Information Security Research Group (COMNETS RG), Universitas Sriwijaya, Indonesia for providing full support for their research necessary.

FUNDING INFORMATION

This work did not receive any funding

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Wisnu Adi Putra	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Deris Stiawan		✓				✓	✓	✓	✓	✓	✓	✓		✓
Qurhanul Rizqie	✓		✓	✓										
Nurul Afifah		✓			✓					✓			✓	
Septiani Kusuma					✓					✓		✓	✓	✓
Ningrum														
Rahmat Budiarto				✓		✓								✓

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nterpretation

R : **R**esources

D : **D**ata Curation

O : **O**riginal Draft

E : **E**diting

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, upon reasonable request.

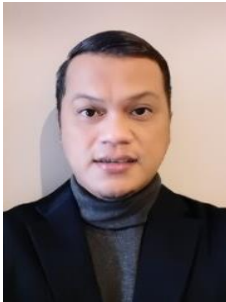
REFERENCES

- [1] K. Inc., "Hacktivists step back giving way to professionals: a look at DDoS in Q3 2022," 2022. <https://www.kaspersky.com/about/press-releases/hacktivists-step-back-giving-way-to-professionals-a-look-at-ddos-in-q3-2022>
- [2] A. B. M. A. Al Islam and T. Sabrina, "Detection of various denial of service and distributed denial of service attacks using RNN ensemble," in *2009 12th International Conference on Computers and Information Technology*, Dec. 2009, pp. 603–608, doi: 10.1109/ICCIT.2009.5407308.
- [3] M. J. Hossain Faruk *et al.*, "Malware detection and prevention using artificial intelligence techniques," *Proc. - 2021 IEEE International Conference on Big Data (Big Data) 2021*, pp. 5369–5377, 2021, doi: 10.1109/BigData52589.2021.9671434.
- [4] K. B. Adedeji, A. M. Abu-Mahfouz, and A. M. Kurien, "DDoS attack and detection methods in internet-enabled networks: concept, research perspectives, and challenges," *Journal of Sensor and Actuator Networks*, vol. 12, no. 4, 2023, doi: 10.3390/jsan12040051.
- [5] D. Alghazzawi, O. Bamasaq, H. Ullah, and M. Z. Asghar, "Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection," *Applied Sciences*, vol. 11, no. 24, 2021, doi: 10.3390/app112411634.
- [6] G. C. Amaizu, C. I. Nwakanma, S. Bhardwaj, J. M. Lee, and D. S. Kim, "Composite and efficient DDoS attack detection framework for B5G networks," *Computer Networks*, vol. 188, 2021, doi: 10.1016/j.comnet.2021.107871.
- [7] A. E. Cil, K. Yildiz, and A. Buldu, "Detection of DDoS attacks with feed forward based deep neural network model," *Expert Systems with Applications*, vol. 169, 2021, doi: 10.1016/j.eswa.2020.114520.
- [8] C. S. Shieh, W. W. Lin, T. T. Nguyen, C. H. Chen, M. F. Horng, and D. Miu, "Detection of unknown ddos attacks with deep learning and gaussian mixture model," *Applied Sciences*, vol. 11, no. 11, 2021, doi: 10.3390/app11115213.
- [9] R. F. Fouladi, O. Ermiş, and E. Anarim, "A Novel Approach for distributed denial of service defense using continuous wavelet transform and convolutional neural network for software-Defined network," *Computers and Security*, vol. 112, 2022, doi: 10.1016/j.cose.2021.102524.
- [10] Y. Wei, J. Jang-Jaccard, F. Sabrina, A. Singh, W. Xu, and S. Camtepe, "AE-MLP: a hybrid deep learning approach for DDoS detection and classification," *IEEE Access*, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3123791.
- [11] L. Zhou, Y. Zhu, T. Zong, and Y. Xiang, "A feature selection-based method for DDoS attack flow classification," *Future Generation Computer Systems*, vol. 132, pp. 67–79, Jul. 2022, doi: 10.1016/j.future.2022.02.006.
- [12] M. V. O. Assis, L. F. Carvalho, J. Lloret, and M. L. Proença, "A GRU deep learning system against attacks in software defined networks," *Journal of Network and Computer Applications*, vol. 177, 2021, doi: 10.1016/j.jnca.2020.102942.
- [13] J. E. Varghese and B. Muniyal, "An efficient IDS framework for DDoS attacks in SDN environment," *IEEE Access*, vol. 9, pp. 69680–69699, 2021, doi: 10.1109/ACCESS.2021.3078065.
- [14] M. Almiani, A. AbuGhazleh, Y. Jararweh, and A. Razaque, "DDoS detection in 5G-enabled IoT networks using deep Kalman backpropagation neural network," *International Journal of Machine Learning and Cybernetics*, 2021, doi: 10.1007/s13042-021-01323-7.
- [15] O. Yousuf and R. N. Mir, "DDoS attack detection in internet of things using recurrent neural network," *Computers and Electrical Engineering*, vol. 101, p. 108034, Jul. 2022, doi: 10.1016/j.compeleceng.2022.108034.
- [16] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *Proceedings - International Carnahan Conference on Security Technology*, 2019, vol. 2019-October, doi: 10.1109/CCST.2019.8888419.
- [17] S. Chormunge and S. Jena, "Correlation based feature selection with clustering for high dimensional data," *Journal of Electrical Systems and Information Technology*, vol. 5, no. 3, pp. 542–549, Dec. 2018, doi: 10.1016/j.jesit.2017.06.004.
- [18] M. A. H. Azmi, C. F. M. Foozy, K. A. M. Sukri, N. A. Abdullah, I. R. A. Hamid, and H. Amnur, "Feature selection approach to detect DDoS attack using machine learning algorithms," *Journal of Informatics and Visualization*, vol. 5, no. 4, p. 395, Dec. 2021, doi: 10.30630/joiv.5.4.734.
- [19] X. Zhao, S. Chen, K. Gao, and L. Luo, "Bidirectional recurrent neural network based on multi-kernel learning support vector machine for transformer fault diagnosis," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 1, 2023, doi: 10.14569/IJACSA.2023.0140114.
- [20] J. F. Toubreau *et al.*, "Capturing spatio-temporal dependencies in the probabilistic forecasting of distribution locational marginal prices," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, 2021, doi: 10.1109/TSG.2020.3047863.
- [21] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," *Applied Soft Computing*, vol. 97, p. 105524, 2020, doi: 10.1016/j.asoc.2019.105524.
- [22] F. Lopes, J. Agnelo, C. A. Teixeira, N. Laranjeiro, and J. Bernardino, "Automating orthogonal defect classification using machine learning algorithms," *Future Generation Computer Systems*, vol. 102, 2020, doi: 10.1016/j.future.2019.09.009.
- [23] M. Kabir, S. Ahmad, M. Iqbal, Z. N. Khan Swati, Z. Liu, and D.-J. Yu, "Improving prediction of extracellular matrix proteins using evolutionary information via a grey system model and asymmetric under-sampling technique," *Chemometrics and Intelligent Laboratory Systems*, vol. 174, pp. 22–32, Mar. 2018, doi: 10.1016/j.chemolab.2018.01.004.
- [24] M. Bach, A. Werner, J. Zywiec, and W. Pluskiewicz, "The study of under- and over-sampling methods' utility in analysis of highly imbalanced data on osteoporosis," *Information Sciences*, vol. 384, pp. 174–190, Apr. 2017, doi: 10.1016/j.ins.2016.09.038.
- [25] J. Ruiz-Bañobre, R. Kandimalla, and A. Goel, "Predictive biomarkers in metastatic colorectal cancer: a systematic review," *JCO Precision Oncology*, no. 3, pp. 1–17, Dec. 2019, doi: 10.1200/PO.18.00260.

BIOGRAPHIES OF AUTHORS



Wisnu Adi Putra    is currently a Master's student at Sriwijaya University. He earned his bachelor's degree at the same university, majoring in Computer Systems. His areas of interest include cryptography, machine learning, and cyber security. He can be contacted at email: wisnuadiputra4@gmail.com.



Deris Stiawan    received the Ph.D. degree in computer science from the Universiti Teknologi Malaysia, Malaysia. He is currently a Professor in Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya. His research interests include computer networks, intrusion detection/prevention systems, and heterogeneous networks. He can be contacted at email: deris@unsri.ac.id.



Qurhanul Rizqie    received bachelor degree in computer science in 2009 from Universitas Sriwijaya, Indonesia, master degree in Electrical Engineering in 2013 from Institut Teknologi Bandung, Indonesia and doctoral degree in biomedical engineering in 2021 from Universiti Teknologi Malaysia, Malaysia. His doctoral research is segmentation of lung area in chest X-Ray image for lung 3D reconstruction. He can be contacted at email: qurhanul.rizqie@ilkom.unsri.ac.id.



Nurul Afifah    received the Bachelor Degree in Computer Engineering and master's degrees in Computer Science at 2014 and 2019. She joined Universitas Sriwijaya as a Lecturer in January 2020. She is currently a Researcher at COMNETS RG. Her research interests include Information Security, IoT system and security, Blockchain and Machine learning. She can be contacted at email: nurul@unsri.ac.id.



Septiani Kusuma Ningrum    graduated from the Faculty of Computer Science, Universitas Sriwijaya. Now, she is researcher at COMNETS RG UNSRI concentration of infosec, especially in the field of lateral movement. She can be contacted at email septianikuning@gmail.com



Rahmat Budiarto    received B.Sc. degree from Bandung Institute of Technology in 1986, M.Eng. and Dr.Eng. in Computer Science from Nagoya Institute of Technology in 1995 and 1998, respectively. Currently, he is a full Professor at College of Computing and Information, Al-Baha University, Saudi Arabia. His research interests include intelligent systems, brain modeling, IPv6, network security, Wireless sensor networks, and MANETs. He can be contacted at email: rahmat@bu.edu.sa.