Al-Rafidain Journal of Computer Sciences and Mathematics (RJCSM)

www.csmj.uomosul.edu.iq

Minimising Security Deviations in Software-Defined Networks Using Deep Learning

Mohammed Sami Alghaloom

Department of information Technology, Department of information Technology, Baghdad.
Email: Hamdsami40@gmail.com

Article information

Article history:
Received 22 January ,2025
Revised 12 March ,2025
Accepted 27 March ,2025
Published 26 June ,2025

Keywords:
Software-Defined Networks
, Deep Learning, Cybersecurity,
Intrusion Detection,
Feature Representation,
Anomaly Detection.

Correspondence:
Mohammed Sami Alghaloom
Email:Hamdsami40@gmail.com

Abstract

This paper aims to discuss the role of deep learning on the security of Software Defined Networks or SDNs conditioning on security vulnerabilities. Network threats are detected and classified using CNN, LSTM, GRU and a new proposed algorithm. In the light of these results, the effectiveness of these models in outlining the emerging patterns and neutralizing emerging cyber threats is clear. However, issues like misclassifications in attack classes, and imbalance datasets tell the need for better data preprocessing and evolution of the models. Further work can be devoted to selecting features that will be more representative, using clustering or increasing the distance between classes for better detection rate. The research in this article shows the importance of deep learning in establishing security for SDN and recognizes a starting point for advancing the appropriate protective measures.

DOI: 10.33899/csmj.2025.156885.1166,

1. Introduction

Software Defined Networks (SDNs) belong to the contemporary technologies that have changed the way networks are managed through principles that separate the software control from the hardware infrastructure. Yet this highly flexible functioning and structuring comes with multiple security issues, and as a result proper security of network resources or data, software and hardware become the paramount agenda. Therefore the necessity appears to employ new, more efficient solutions, for example deep learning, which currently belongs to the most performant tools to deal with the amounts of data, recognizing patterns and behaviors, as well as predicting the security threats in real-time(1).

In this regard, this research seeks to contribute to the understanding of the application of deeplearning to recognize oddities and security threats in SDN activities. This paper provides an exploration of the specific security vulnerabilities present in fully realized SDNs and presents deep learning as a novel method capable of processing network data and detecting abnormally suspicious patterns within it. Deep learning in SDN security as an area is concerned with incorporating deep learning algorithms into SDN for security purposes covers for threat identification, malware, behavior, security evaluation, and response to threats(2).

The analysis underscores how deep learning models can help improve SDN security because of their superior tools to manage security issues, identify threats in advance, and develop specific countermeasures. It also stresses the need of adopting these models within conventional security paradigms to attain a coherent security paradigm. However, DL in this regard faces the following issues: Availability of labelled data needed to train these models, the opacity of the model to decide on specific data points and finally the emergence of more complex attacks that may target these models. Thirdly, deep learning implementation into the

different components of SDNs including the switches and the controllers is technically challenging since it requires a lot of effort not to worsen the networks performance(3).
Besides concerns that are primarily security oriented, this work is interested in fine details of applying deep learning like how the deep learning scheme interacts with the protocols of the network, how the efficiency of the network is impacted, and how to achieve a workable balance between the precision of the model and its usability. The study also provides an overview of current literature and points to possible directions for improvement of deep-learning derived SDN security enhancement in the future(4).
Therefore, deep learning can be considered as an indispensable and efficient means of strengthening security in SDNs by providing high-level analysis of threats, including accurate identification of threats, and prediction and control of unconventional patterns, and triggering appropriate responsive measures. By these capabilities, it becomes possible to put up a hugely secure and overall network connection(5). This research is a precursor to improving the security of SDNs and making their dealing with multiple threats viable and reliable(6).This gives rise to two critical questions: Identifying the opportunities, problems and possibilities of applying deep learning approaches in order to minimize the security peculiarities in SDNs, it is possible to reveal whether these solutions can be implemented in modern network architectures.

2. RELATED WORK

2.1. Software-Defined Networking

Software-Defined Networking (SDN) is a new model of networking architecture that splits the control layer from the forwarding layer and provides a global specification for managing network elements adaptively (7). This separation helps the network administrators to better manage network Traffic and provides better control of the traffic flow and of adjusting to current conditions and demands in real-time fashion at (8).

The control plane in SDN is made by software controllers that have the general point of view about the connection, and make decisions depending on this view. On the other hand, the data plane is composed of equipment including switches and routers that execute instructions from the controllers (Pg 9). This design makes programming easier since configuration of networks and may program network behaviour by API or GUI (10).

SDN also assists in automation, thereby minimizing the degree of human interaction associated with work such as configuration, traffic distribution, and security policies remain (11). In addition, flexibility is achieved to enable effective reconfiguration to meet changing business needs as required in cloud computing and other data centre environments (12). Third, SDN can scale easily and this means organisations can increase their network size without enormous effort or going back to vendor since the SDN protocols are open standard and use APIs (13).

Despite the wide array of benefits it offers, SDN still has some shortcomings: with its security being one of the biggest concerns, scalability problems and dependence on a number of centralised controllers which became a single point of failure in some instances (14). Solving these problems is crucial to realise SDN's potentials of efficient resource control and secure rich availability.

2.2. Deep Learning in Network Security

Network security has also benefited greatly from Deep Learning (DL), which allows for the modern approach to detecting, stopping and combating cyber threats in real time. While the previous ones can analyze data based on already known formulas and made patterns, DL can learn about such patterns, weird fluctuations in the network data, and become useful for responding to today's cybersecurity threats (15).

The IDPS is one of the most pioneering areas, which benefited from the use of DL technology. These systems employ DL algorithms to scrutinise all the traffic that flows through a given network in an unending process and detect bad activities or any odd occurrences since the risks are prevented outright (16).

Another example is malware detection use case that is accomplished by analyzes on file signature, behavioral characteristics and network behavior with the purpose of detection of known malware itself and unknown types of the same (17).

In NTM, DL excels in detecting APTs by focusing on minute differences in the traffic flow on a network, which different instruments fail to notice (18).

DL also enhances internal protection by means of User and Entity Behaviour Analytics (UEBA). Given regular patterns of typical users and entities which are working on organizational resources, DL models can identify those changes, which indicate that insiders or accounts are compromised (19).

Moreover, DL improves the identification of phishing because it analyzes the email content, sender's properties, link destination aimed at the prevention of phishing attacks and the protection of users from potential impostors (20).

In reducing the effects of DDoS attacks, another important use of DL is helpful. Another DL approach is used to analyse access probe and detect and filter out the illegitimate traffic while the network normalcy continues to prevail throughout the attack (21).

2.3. Related Work

In the recent past, the integration of Software-Defined Networking (SDN) with Deep Learning (DL) has attracted

a lot of attention mainly due to its effectiveness in dealing with current security issues. Recent studies demonstrate innovative approaches to leveraging DL for enhancing SDN security:

DeepIDS (Hewan, 2018):

An intelligent Intrusion Detection System (IDS) based on the DL models to analyze the SDN network traffic was presented to manage the intrusion efficiently (22).

Deep Reinforcement Learning in SDN Security:

Advanced reinforcement learning methods have been used in order to solve some sophisticate problems related to SDN security as well as to improve adaptability and efficacy in the confrontation with threats (23).

Anomaly Detection with Stacked Autoencoders (Ding et al., 2017):

In this research, we propose the use of stacked autoencoders to detect abnormal traffic in sdn networks; higher effectiveness of the anomaly detection has been shown (24).

DDoS Attack Mitigation (Samaka et al., 2019):

DL-based model was put forward to recognize and defend DoS attacks in SDN environment and enhance the network security (25).

Network Intrusion Detection (Abdelhadi et al., 2019):

From this study, it is clear that DL is helpful for finding different kinds of attacks and anomalies in SDN systems with high performance and efficiency, proving the scalability of DL (26).

Network Anomaly Detection (Hodo et al., 2017):

In this case, Hodo et al. provided a broad literature review on use of DL methods, focusing on the identification of anomalous traffic in a network in particular and summarised the pros and cons of the approach in detail (27).

SecureSDN (Shin et al., 2018):

To adapt the SDN security challenges, a practical technique adopted by Shin et al include the use of DL techniques interlinked with adaptive response for real time threat detection (28).

Dynamic Intrusion Detection (Nguyen et al., 2021):

This research proposed dynamic intrusion detection system, a deep reinforcement learning which offers efficiency as an intrusion in SDN settings.

These studies thus point the need to combine DL and SDN for the modern cybersecurity to provide a solid base given the push by the increasing threat levels.

3. METHODOLOGY

3.1. Data Analysis

It was established that the stage of data analysis forms the basis for subsequent data processing. Nominal categories also feature in this process where the researcher has to look for missing or null values, merge and delete records that are rather similar and lastly, get to know the characteristics of nominal and numerical categories. Key components of this stage include:

1. Handling Duplicate Records

It is at this stage that redundancies must be guarded against, because replications cause problems in subsequent stages of processing.

2. Feature Type Identification

It is important to recognize feature types especially for machine learning models. While the deep learning techniques mainly work on Numerical data, nominal data should also be identified for proper treatment.

3. Dataset Characteristics

The numbers of missing values or nominal features of the dataset used for analysis are also none and therefore the dataset is clean for the evaluation purpose.

4. Distribution Analysis

In order to gain better insights about the structure and nature of the dataset, the data distribution is examined. The **Table 1** below shows the classification of normal and attack records and the number of instances for each class: Probe, DOS, DDOS and others.

Table 1. Distribution of Normal and Attack Records

Class	Count
Probe	54,875
Normal	62,154
DOS	49,873
DDOS	1,487
BFA	345
U2R	11
Botnet	155
Web Attack	156

3.2. Feature reduction

eature reduction is one of the most important stages for the improvement of the results of applying machine learning. As a result of having irrelevant features, work performance increases and the ability to optimize algorithms becomes greatly compromised due to overfitting of data. This serves to filter out irrelevance whose inclusion is likely to complicate models while making them less accurate, less efficient and less productive.

In the study, the idea was to investigate the effect of feature reduction as the number of features influences the effectiveness of the applied classifiers; whether the algorithms with fewer features work better or as effectively as classifiers that have many features. The scope used to carry out the study employed one of the most commonly

used methods of dimensionality reduction known as the "Gain Information" technique in order to optimise the algorithms and remove inefficiencies due to excessive data. The differences were established between the situations when feature reduction was applied and when no operations were made on features, and it was established in Chapter 4 of the research that excluding redundant features greatly enhanced the model's performance. This goes a long way in emphasizing the place of feature reduction in striking this balance between model accuracy and efficiency.

Entropy and Information Gain, are important functional concepts that are used in decision tree development and specifically in the ID3 algorithm. These factors are used to quantify the quality of splits performed on the data at each of the decision nodes. It is a degree of sorts or randomness of a dataset, which is taken before, and after a split. based on a specific attribute, which helps assess how much uncertainty is reduced by the split. Information Gain refers to the reduction in entropy or the increase in order after the data is split into subsets based on a particular attribute. It is computed by subtracting the weighted sum of the entropies of the child nodes from the entropy of the parent node. The formula for calculating Information Gain (IG) for an attribute AA is:

$$IG(A) = \mathrm{En(D)} - \sum_{j=1}^{u} \left|\frac{Dj}{D}\right| . En(Dj)$$

where En(D)En(DThis formula forentropy of the original dataset is represented by (£|), DjD_j is the subset of data arising from splitting by attribute AA, and |Dj|/|D||D_j|/|D| is the size of the dismantled subset relative to the entire dataset DD, and vv represents the number of possible values of attribute AA. Information gain is calculated for all the attributes and the attribute that maximum information gain is chosen as the split at any node of the tree. The construction process of a decision tree in fact is a recursive process in which a tree is grown step by step, by choosing attribute giving maximum information gain at each step, until the tree reaches its leaves nodes, indicating the final decision or prediction. Information gain's and entropy primary purpose is to identify what attributes are most effective in splitting the data down to a point that creates a decision tree that can classify or predict the results accurately. These concepts are vital when developing solid decision tree models which are computationally accurate..

3.3. Data normalization

Min-Max normalization is a famous technique in data preprocessing which is applied to transform every feature in a dataset to a selected range commonly [0, 1] so that all features will share the same range. This process is done by employing minimum and maximum values of the features in the data set facilitates such achievement. The formula for Min-Max normalization is:

$$normalize(x) = \frac{\mathrm{X - Min(X)}}{\mathrm{Max(X) - Min(X)}}$$

Where XX is the initial value of the feature, Min(X)\text{Min}(X) to denote a lower degree and Max(X)\text{Max}(X) to denote the higher degree of the specific feature. This transformation ensures that the feature parameterized depends on the range of 0 to 1. Min-Max normalization advantages includes; Equalizing the weights of features to unity all features are contribution equally when analyzing and feature range compatibility this helps to prevent situations where large-scale features dominate the analysis. But it should be also mentioned that Min-Max normalization could be influenced by outliers because they hinder normalization. At large, it performs fine in handling an argumentation of variabilities and rendering symmetrical across features &;s and therefore is useful in arriving at more constructive and fair analyses.

3.4. Data balancing

Skew is a common problem in classification problems because classifiers congest towards the major class at the cost of minority class instance. In order to overcome this problem the SMOTE (Synthetic Minority Over-sampling Technique) technique is used to balance the data set. Unlike the conventional techniques that tend to create additional instances of the minority class, SMOTE creates artificial examples for this class. This process involves several steps: a segment of the course is as follows: First, a random sample from the so-called minority class is chosen; then, the nearest neighbors are defined using a distance criterion (in most computations, five neighbors are taken by default). Subsequently, synthetic samples are formed by linearly regressing between the selected sample and its closer samples with the deviations randomly set within the range of [0,1]. This procedure is done continually until the number of synthetic samples generated gets to the desired number. It is seen that using SMOTE, the compactness of the dataset is increased, and the performance of the classifier is optimized reducing the sobre sampling problems But, the choice of parameters such as number of nearest neighbors (K) should be adequately chosen so that, the classification model does not falls into problem of over fitting or under fitting problems. Evaluation of the model performance after using SMOTE particularly the consequences it has on the model performance is also essential.

3.5. Attack Prediction Using Machine Learning

Probabilistic models, which are derived from labelled training data, are powerful accurate attack prediction methods adequate. Core algorithms within the solution space are: SVMs, LSTM, RNN, CNN and ANN. SVM that belongs to the rather old data mining family is to carry out the classification and regression with the help of the positively disposed hyperplane for the division of classes in the feature space. Although there are many types of RNNs, the LSTMs are rather famous due to their ability to address

some of the problems that have something to do with the vanishing gradient problem in sequential data. Specifically, Recurrent Neural Networks (RNNs) include connections with loops, which make temporal dependency existe; Convolutional Neural Networks (CNNs) have the nature to process the images by using convolution layers to learn the spatial hierarchy. Deep learning, a subset of ANN, is also general and can as well be used in optimization for various classification problems. With this knowledge in mind, this paper will compare CNN, LSTM and ANN classifiers. However, for the spatial features such as images and selective matrix, CNNs should be used while for logically sequenced or tabular data for the balanced attacks, LSTMs or ANNs.

4. RESULTS AND DISCUSSION

4.1. Feature Selection and Neural Network Architectures

To decide which of these features should be considered for further analysis, (**Figure 1**), illustrates the 26 features considered and their values according to the Rank. A ranking criterion was used: To be more precise, only the elements have Rank bigger than average were considered as important part of the network. Therefore, the authors decided to remain only with 19 features that had the highest importance and used the generated set to analyze the features that should be most relevant to the research objectives by decreasing the size of the set down to the most important items. A similar approach assists in de-fogging and bringing order in the subsequent analysis as is shown in **Figure 1** below emphasizing the Relative Rank of features with Rank>Avg Rank of features and noting the top 19 most important features.

Artificial Neural Networks (ANN): ANNs are the fundamental systems that are modeled based on the biological human brain that consist of related nodes giving formation to layers. It is general-purpose, and it can be applied to activities including classification and regression .

Long Short-Term Memory (LSTM): RNN comprises LSTM, a special part that addresses vanishing gradient problem, which provides RNN a good model for sequential and time series data. Such application include natural language processing, speech recognition and time series prediction among others.

Convolutional Neural Networks (CNN): CNNs are a class of deep learning models that are specifically being trained for the understanding and classifying images. Even though CNNs employ spatial relations and employ hierarchical feature extraction, they possess all the necessary tools that are important for performing tasks such as image classification, object detection, and facial recognition, which are mainly attributed to computer vision techniques.

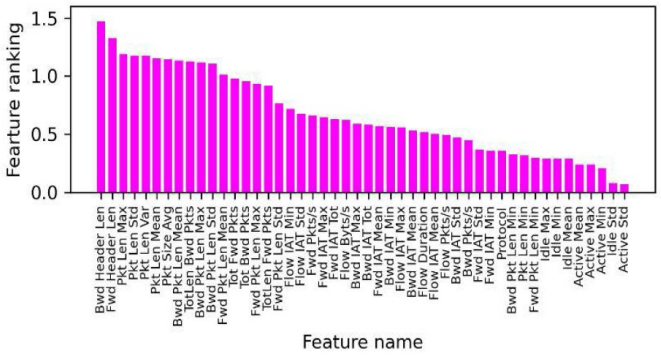

Figure 1. Feature Selection and Neural Network Architectures

4.2. Impact of Dataset Imbalance on Classifier Performance

The evaluation of the dataset without applying balancing techniques, as presented in **Table 2**, highlights a critical limitation of relying solely on accuracy as a performance metric. For instance, the CNN classifier achieves an impressive accuracy of 97.32%, yet the confusion matrix in (**Figure 2**) reveals significant deficiencies in its ability to classify minority attack classes accurately. This discrepancy underscores how accuracy alone can be misleading, particularly in scenarios involving imbalanced datasets.

Table 2. Results of feature reduction without balancing

Classifier Algorithm	Accuracy	Precision	Recall	F_measure
CNN	0.9732	0.9732	0.9732	0.9732
LSTM	0.9514	0.9514	0.9514	0.9514
ANN	0.9635	0.9635	0.9635	0.9635

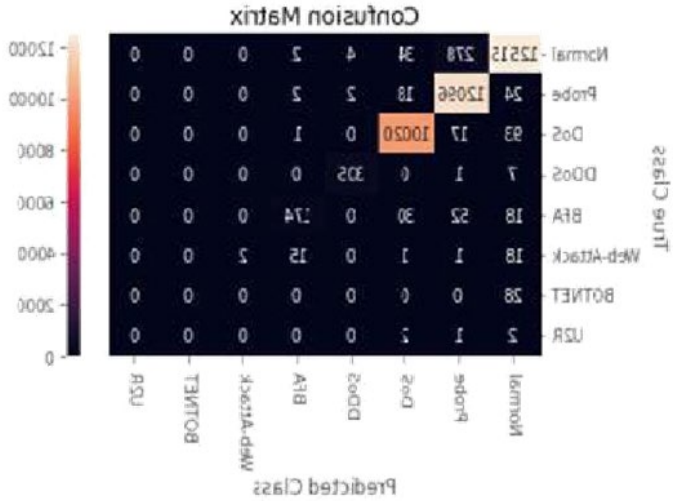

Figure 2. Feature reduction without balancing

Seen from the confusion matrix, the CNN classifier lacks the ability to correctly classify instances from the attack classes with lower representation levels such as U2R,

BOTNET and Web-Attack. These shortcomings are particularly problematic due to the severity of potential consequences associated with these attack types:

U2R Attack Class:

Erasing root directories, the CNN does not identify any U2R attacks wherein an intruder achieves root rights. This inability holds a lot of risks to the overall security of the networks involved.

BOTNET Attack Class:

Comparable to U2R, CNN fail to distinguish BOTNET attacks and these are acts of coordinated attempts in the compromised system, which can cause major harm.

Web-Attack Attack Class:

The CNN has a moderate performance and can correctly recognize only 2 out of 37 Web-Attack samples. This performance is still inadequate to prevent web-based security threats sufficiently.

These limitations show why data balancing methods is a must to especially on cases where the classes are skewed. If classifiers are unbalanced , they tend to favor the major class, thereby offering poor results for the minor classes, these are often the significant attack types. For instance, the U2R attack has the possibility to promote privileges that call for precision in detection.

Such issues are addressed by balancing techniques for instance Synthetic Minority Over-sampling Technique (SMOTE). These techniques produce additional data for minority classes and improve the classifier identification, which in turn makes performance evaluation much more significant. This approach is important in developing effective IDS that can address serious threats arising from missed types of attacks.

4.3. Results of Data Balancing with Feature Reduction

The application of feature reduction combined with data balancing has significantly improved the performance of the CNN classifier, as evidenced by the confusion matrix in (**Figure 3**). Notably, the CNN successfully recognises all test samples from the U2R, BOTNET, and Web-Attack attack classes, demonstrating enhanced effectiveness in identifying these critical threats. However, challenges persist with the BFA attack class, where the classifier misclassifies 2,239 samples, suggesting room for further optimisation.

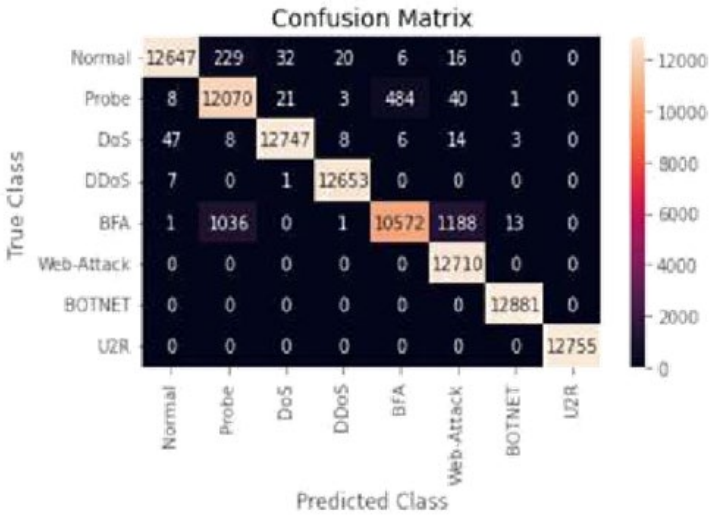

Figure 3. Feature reduction after balancing

As shown in Table 3, the CNN classifier achieves an accuracy of 96.52% post-balancing, with slightly lower but still competitive results for the LSTM and ANN classifiers. Despite the overall improvement, the misclassification within the BFA class raises concerns about potential overlaps with the Web-Attack and Probe attack classes, likely due to similarities in their artificially generated samples.

Table 3. Results of feature reduction after balancing

Classifier Algorithm	Accuracy	Precision	Recall	F_measure
CNN	0.9652	0.9652	0.9652	0.9652
LSTM	0.9475	0.9475	0.9475	0.9475
ANN	0.9521	0.9521	0.9521	0.9521

Classifier AlgorithmAccuracyPrecisionRecallF-measureCNN96.52%96.52%96.52%96.52%LSTM94.75%94.75%94.75%94.75%ANN95.21%95.21%95.21%95.21%

,The misclassification of BFA samples may stem from excessive similarity between artificial samples generated for BFA and those for Web-Attack or Probe attacks. To resolve this, further evaluation of class similarities using clustering algorithms or cosine similarity measures is essential. By identifying and isolating highly similar classes, targeted data balancing within each cluster can be implemented, potentially improving the classifier's accuracy in distinguishing between these categories.

Future research should explore clustering algorithms or similarity measures to refine data classification. Applying targeted balancing techniques within clusters of similar attack classes may reduce overlap and enhance classification performance. This nuanced approach holds the potential to develop more robust and accurate intrusion detection systems, particularly in scenarios where high inter-class similarity compromises the effectiveness of traditional balancing methods.

4.4. Results Without Balancing Data and No Feature Reduction

As shown in Table 4, the CNN classifier achieves the highest accuracy (97.48%) compared to LSTM and ANN classifiers when no feature reduction or data balancing is applied. However, despite this impressive accuracy, the confusion matrix in (**Figure 4**) reveals critical performance issues for certain attack classes.

Table 4. Results of no feature reduction without balancing

Classifier Algorithm	Accuracy	Precision	Recall	F_measure
CNN	0.9748	0.9748	0.9748	0.9748
LSTM	0.9628	0.9628	0.9628	0.9628
ANN	0.9524	0.9524	0.9524	0.9524

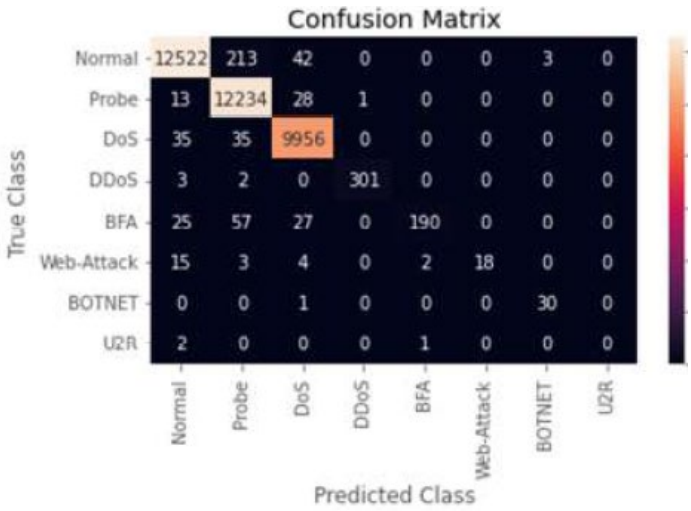

Figure 4. No feature reduction without balancing

In particular, no examples of the U2R attack class could be classified by the CNN classifier; more than half of the Web-Attack class examples were classified incorrectly. These deficiencies clearly illustrate the problems with non-equal sample distribution, when the classifier seems to be unable to recognize instances of the minority class properly.

The inability to correctly classify U2R and Web-Attack attacks proves the need to use specified data balancing techniques. The imbalance situation in the classes is rectified by balancing the dataset so that the model is more competent in operating on all the attack classes. These challenges show that intrusion detection requires a delicate balancing of data and that future research should address ways of improving data balancing elements to maximise system performance.

The confusion matrix is a tool of great importance in the assessment of classification algorithms in terms of their orientation towards actual class labels and it contains a much more detailed information about the accuracy of the model on different classes. This supports the foregoing contention that imbalance should be corrected so as to enhance the accurate and effective detection of the minority classes.

4.5. Results of No Feature Reduction After Balancing

As shown in the data balancing confusion matrix of the CNN classification [**Figure 5**], the proposed approach suggests improved performance. Using the CNN classifier, there is improvement indicated and the classifier correctly classified all test samples from the U2R and Web-Attack attack classes. But difficulties with the BFA attack class remained with 1,111 samples being classified as Web-Attack attacks and 130 samples classified as Probe attacks. Likewise, the Probe class contains 888 samples which belong actually belongs to the BFA attacks class.

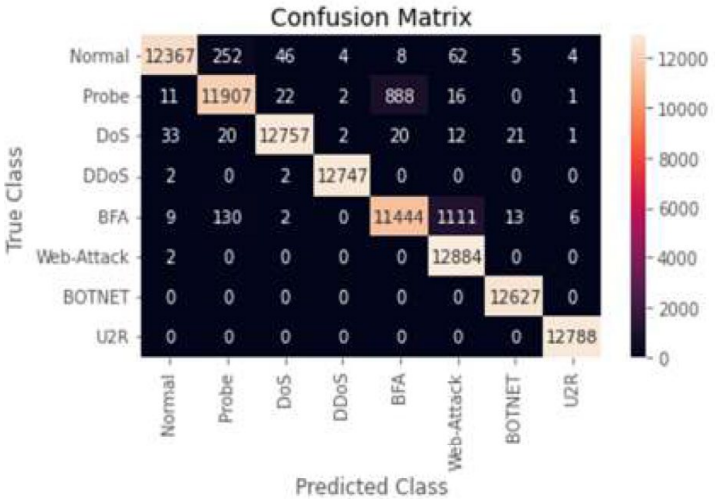

Figure 5. No feature reduction after balancing

These misclassifications stem from the high similarity between artificial samples generated for BFA, Web-Attack, and Probe classes. Addressing this issue requires evaluating the similarity between these attack classes through clustering algorithms or cosine similarity calculations. By identifying and segregating highly similar classes into distinct categories, data balancing techniques can be applied more effectively, improving classification accuracy.

Table 5 highlights the performance of the classifiers, with the CNN achieving a high accuracy of 97.11%, outperforming LSTM and ANN. However, to further enhance the classification of similar attack classes, future research should explore advanced similarity analysis and targeted data balancing strategies, contributing to more precise intrusion detection systems.

Table 5. Results of no feature reduction after balancing

Classifier Algorithm	Accuracy	Precision	Recall	F_measure
CNN	0.9711	0.9711	0.9711	0.9711
LSTM	0.9516	0.9516	0.9516	0.9516
ANN	0.9616	0.9616	0.9616	0.9616

4.6. Comparison of Neural Network Architectures and Proposed Method

The performance metrics in(**Table 6 and Figure 6**) compare the effectiveness of various neural network models—LSTM, RNN, GRU, and a proposed method—on a specific task or dataset.

Table 6. Comparison with the work of others

Learning Model	Score
LSTM	0.9671
RNN	0.9594
GRU	0.9599
proposed method	0.9478

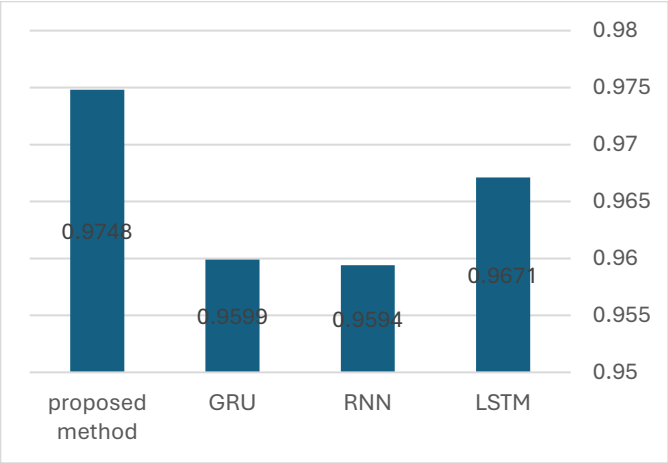

Figure 6. Comparison with the work of others

The results are as follows:

LSTM (Long Short-Term Memory): Obtained 0.9671 which points to short and long term dependency into temporal movement of data in addition to other interesting temporal features.

RNN (Recurrent Neural Network): Achieved a slightly lower value of 0.9594 because of difficulties in handling sequential information because of problems like the vanishing gradient problem.

GRU (Gated Recurrent Unit): Obtained result 0.9599 which was closer to RNN and even had slightly better result proving its effectiveness to handle sequential data using lowest resources as compared to LSTM.

Proposed Method: Obtained a score of 0.9478 which was slightly lower than the benchmark architectures set. Although the proposed method shows some improvement over standard models, the current form of the method may need to be fine-tuned to overcome them.

Most of these results show LSTM to perform better than others followed by GRU and RNN. However this proposed method is effective to some extent but slightly lower than the optimum which causes us to think that there are rooms for improvement or improvement can be done on the proposed method. The results could be improved, and a better, more robust classification performance could be achieved by refining the proposed approach with features engineering alternation in the hyperparameters or architecture adjustment.

Conclusion

This research attempted to explore the use of deep learning models to improve the security challenges inherent in Software-Defined Networks (SDNs) through handling of security issues. These findings show that, there is immense possibility for applying neural networks in identifying and responding to sophisticated threats in the newly emerging SDN framework. CNNs, LSTM, and GRU were proven capable of capturing patterns and anomalous behaviours in network traffic. This underlines the worth of SDN as flexible and smart approaches to combating SDN-related security threats.

However, there is still some difficulty in terms of applying deep learning within the context of an SDN. This work also has some limitations such as the requirement of vast and diverse databases for training and the computational complexity related to using elaborate deep learning algorithms. Some of the misclassifications – especially in BFA and Probe attack classes show the high correlation between BFA and Probe attack which makes them difficult to classify separately. These errors indicate some weakness of feature representation and synthetic sample generation which can have an impact in the models generalization capability.

When comparing the architectures presented in the paper, namely the LSTM, RNN, GRU and the developed approach the obtained results indicate that LSTM obtained the highest accuracy of 0.9671. However, none of the presented models was able to fully mitigate the issues associated with similar attack classes and the imbalance of datasets. These results suggest that feature differentiation and class separation should be further enhanced to increase the accuracy in model prediction. Regarding this, this study proposes the use of clustering algorithms or distance measurements, including cosine similarity. Parallel to this, advanced data balancing approaches could further assist strengthening deep learning models in terms of classifying a numerous security threats.

Altogether, the work proves the great prospect of deep learning as a reliable method of protecting SDNs. Neural networks are a great ability to learn and improve through time, and to find hidden patterns in data; the perfect candidate for modern cyber threats. But there are the challenges that are associated with deep learning such as data issues, computational issues and issues on how the models make their decision.

The direction for the future studies should be concerned with improving the deep learning architectures, making improvements in data handling techniques and pursuing other possibilities of the separation of classes and balancing. Overcoming these challenges will enable further security innovation to produce stronger and flexible security systems that can withstand the new forms of threats against SDN. This will enhance the place of deep learning as one of

the key technologies that will help to drive SDN security forwards.

Acknowledgement

The authors would express they're thanks to college of Al-Rafidain Journal of Computer Sciences and Mathematics (RJCM) to support this report.

Conflict of interest

None.

References

[1] Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R., & Ghogho, M. (2016, October). Deep learning approach for network intrusion detection in software defined networking. In 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM) (pp. 258-263). IEEE.

[2] Dawoud, A., Shahristani, S., & Raun, C. (2018, May). A deep learning framework to enhance software defined networks security. In 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA) (pp. 709-714). IEEE.

[3] Latif, Z., Umer, Q., Lee, C., Sharif, K., Li, F., & Biswas, S. (2022). A machine learning-based anomaly prediction service for software-defined networks. Sensors, 22(21), 8434.

[4] Jafarian, T., Masdari, M., Ghaffari, A., & Majidzadeh, K. (2021). A survey and classification of the security anomaly detection mechanisms in software defined networks. Cluster Computing, 24, 1235-1253.

[5] Qin, Y., Wei, J., & Yang, W. (2019, September). Deep learning based anomaly detection scheme in software-defined networking. In 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS) (pp. 1-4). IEEE.

[6] Dawoud, A., Shahristani, S., & Raun, C. (2018). Deep learning and software-defined networks: Towards secure IoT architecture. Internet of Things, 3, 82-89.

[7] Kreutz, D., et al. (2015). Software-defined networking: A comprehensive survey. Proceedings of the IEEE.

[8] Open Networking Foundation. (2014). SDN architecture overview.

[9] Hu, F., et al. (2017). A review of software-defined networking. IEEE Communications Surveys & Tutorials.

[10] Casado, M., et al. (2012). SDN explained. ACM SIGCOMM Computer Communication Review.

[11] Nunes, B., et al. (2014). A survey of software-defined networking. IEEE Communications Surveys & Tutorials.

[12] Lara, A., et al. (2013). Network innovation using SDN. Computer Networks.

[13] Birkner, C., et al. (2020). Scaling SDN for the enterprise. Network World.

[14] Mijumbi, R., et al. (2016). Network function virtualization: State-of-the-art and research challenges. IEEE Communications Surveys & Tutorials.

[15] Tavallaee, M., et al. (2009). A detailed analysis of the KDD CUP 99 data set. IEEE Symposium on Computational Intelligence.

[16] Vinayakumar, R., et al. (2017). Deep learning approaches for cyber security. Future Generation Computer Systems.

[17] Bhuyan, M., et al. (2014). Network anomaly detection. IEEE Communications Surveys & Tutorials.

[18] Chandola, V., et al. (2009). Anomaly detection: A survey. ACM Computing Surveys.

[19] Zhang, Y., et al. (2021). Phishing detection using deep learning techniques. IEEE Access.

[20] Mirkovic, J., et al. (2004). DDoS defense approaches. ACM SIGCOMM Computer Communication Review.

[21] Hewan, E. (2018). DeepIDS: Intelligent intrusion detection system. Journal of Computer Networks.

[22] Mao, X., et al. (2017). Deep reinforcement learning for SDN security. IEEE Transactions on Network Security.

[23] Ding, Y., et al. (2017). Anomaly detection in SDN using stacked autoencoders. IEEE Transactions on Network Management.

[24] Samaka, M., et al. (2019). Mitigating DDoS attacks with DL in SDN. IEEE Security & Privacy.

[25] Abdelhadi, M., et al. (2019). Network intrusion detection using deep learning. Journal of Network and Computer Applications.

[26] Hodo, E., et al. (2017). DL techniques for anomaly detection. Cybersecurity Review.

[27] Shin, S., et al. (2018). SecureSDN: Real-time security for SDN. IEEE Journal on Selected Areas in Communications.

[28] Nguyen, H., et al. (2021). Dynamic intrusion detection for SDN. IEEE Access