Sigma J Eng Nat Sci, Vol. 44, No. 1, pp. 388–403, February, 2026

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2026.1988

Research Article

Enhancing bidirectional gated recurrent unit with activation mechanism for anomaly classification for network security

N. Senthil MADASAMY*, A. Noble Mary JULIET, P. Boopathi RAJAN

[1]*Dr. Mahalingam College of Engineering and Technology, Udumalai Road Pollachi, 642003, India*

ARTICLE INFO

Article history
Received: 22 July 2024
Revised: 03 October 2024
Accepted: 09 January 2025

Keywords:
Anomaly Classification; Bidirectional Gated Recurrent Unit and Activation Function; Deep Learning; Network Security; NSL-KDD Dataset; UNSWW-NB 15

ABSTRACT

In a digital world, cyber-attacks are increasingly common, raising concerns that existing anomaly detection models might not effectively handle intricate threat scenarios. As the demands for network systems grow. Historically, the update and reset gates of the Gated Recurrent Unit (GRU) designed to controls flow in input data across time steps have faced challenges in identifying anomalies in security monitoring, traffic log analyses, and packet flow assessments. To address anomalies, reduce time expenditure, and improve network security, the proposed research utilizes a Deep Learning (DL) technique named Structured Activation Module Loop Framework Unit and an efficient activation module unit, which integrates a Bidirectional Gated Recurrent Unit (Bi-GRU) model that includes update and reset gates for controlling information flow on the basis of the classification context of the input data. The suggested structured activated loop framework monitors error data straight in the update gate without demanding the reset gate, enabling several checks in a loop format. The activation module unit precisely divides the class data to predict the appropriate output characteristics for resolving missing values. This utilizes network intrusion datasets (UNSW-NB15) and Neural Simulation Language- Knowledge Discovery in Databases (NSL-KDD), both commonly used for (NID) Network Intrusion Detection systems, along with pre-processing data and assessments for data splitting through training the model and testing procedures. Similarly, the proposed performance is assessed using different metrics such as F1-meaures, recall, value of precision, accuracy, with overall accuracy to evaluate the efficiency of the suggested deep learning study. The results obtained precisions of 0.99, 0.97, 0.97, and 0.97 in NSL-KDD, 0.98, 0.99, 0.98, and 0.98 in UNSW-NB 15. Still, the assessment of recommended models demonstrates the efficacy of the research. The present research attempts for find and improve creation of detection of anomaly models for network and cyber security regarding hackers and malicious attacks.

Cite this article as: Madasamy NS, Juliet ANM, Rajan PB. Enhancing bidirectional gated recurrent unit with activation mechanism for anomaly classification for network security. Sigma J Eng Nat Sci 2026;44(1):388–403.

***Corresponding author.**
*E-mail address: senkav1293@gmail.com, senthilmadasamy@drmcet.ac.in

This paper was recommended for publication in revised form by Editor-in-Chief Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey
© Author. This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

INTRODUCTION

In this era of technology, both businesses and individuals are equally worried about network security. Quick expansion of network and the increasing dependence on interacted systems led to a rise in potential cyber threats. Identifying and categorizing abnormal network activities that differ from normal behaviour is a cricial element of network safety [1]. Anomaly classification involves recognizing and grouping atypical network behaviours. These abnormalities may suggest possible security risks, such as malware infections, illegal access attempts, or breaches. Nevertheless, training these models typically requires manual techniques for labelling and categorizing anomalies, leading to potential restrictions and disadvantages. An important disadvantage of manual techniques for anomaly classification is the subjectivity, time-consuming nature, labour-intensive process, and possibility of human error [2]. In recent times, researchers have been employing technology based on Artificial Intelligence (AI) for anomaly classification [3]. Benefits of Deep Learning (DL) and Machine Learning (ML) offer many uses network security [4]. Anomaly classification utilizes ML and DL techniques to examine network traffic, detecting deviations from the usual patterns and categorizing correct and incorrect signals [5]. Different approaches have been experimented with to detect anomalies in network structures using the backing of ML and DL technologies. Likewise, lack of strong authentication can enable attackers to examine and intercept the traffic [6]. Even with all its capabilities and attributes, network security remains a major worry.

Similarly, various traditional models have tried to classify anomalies. As an example, the traditional approach has used a successful anomaly detection system that leverages Mutual Information (MI) and incorporates a Deep Neural Network (DNN) for IoT network. The evaluation used the IoT-Botnet 2020 dataset and attained improved exactness [7]. In the same way, the prevailing model presented an integration of SMOTE over sampling method and 1-D CNN method [8] were the statistical analysis and secured robustness provide better results [9] for minority classes of attacks classification. The testing process has made use of the UNSW-NB15 dataset. The results indicated the existing research has achieved superior values on assessment metrics [10]. Similarly, a DL technique has been utilized to detect different kinds of anomalies in IoT data flow. It was assessed using five different datasets and achieved improved outcomes [11], utilizing Mogrifier Gated Recurrent Unit (GRU) and Multi Scale Convolutional Neural Network (MSCNN) for log-based anomaly detection to differentiate between usual and unusual logs by examining local and global dependencies [12] resulting in high precision statistical classification of severity levels [13]. Similarly, the current model has integrated both CNN and GRU model [14] for identifying anomalies in encrypted network traffic. It has utilized 3 publicly available datasets such as NSL-KDD, CIC-IDS-2017 and UNSW-NB15. The existing model's experiment findings revealed accuracies of 93.10%, 91.21% and 90.17% on NSL-KDD, CIC-IDS-2017 and UNSW-NB15 dataset respectively [15]. Likewise, previous models such as the Mountaineering Team-Based Optimization (MTBO) algorithm, the AWID3 dataset, and the Technical and Vocational Education and Training-Based Optimizer (TVETBO) optimizer have been employed to improve robustness and address challenges with anomalies [16]. The increase in efficiency and robustness has been a positive outcome [17]. Similarly, a traditional approach has been used to create an IDS for IoT that relies on anomalies. Afterwards, CNN has been used in 1D, 2D, and 3D. It has been tested on four specific datasets - IoT Network Intrusion, BoT-IoT, IoT-23 intrusion detection, and MQTT-IoT-IDS2020 dataset [18] achieving improved accuracy and network security [3, 19]. Consequently, the traditional models have achieved successful outcomes without introducing new elements [20, 21].

The study on supervised deep learning models for detecting network traffic abnormalities early on points out a number of ongoing obstacles that hinder the success of current approaches. One main problem is the absence of standardized reference markers, making performance evaluation difficult and raising the risk of over fitting. This issue has made worse by a large amount of wrong identifications due to using possibly compromised or outdated data, which makes it challenging to distinguish between regular and abnormal traffic patterns. Moreover, when datasets increase in size, they frequently encounter more noise, causing models to have difficulty adjusting to new attack strategies, especially when dealing with unfamiliar anomalies. The issue of unequal distribution of classes in datasets remains, affecting the model's learning of minority classes. Unlike previous studies, the propounded model includes enhancements to tackle these issues. Using both UNSW-NB15 and NSL-KDD set of data, the model integrates a Structured Activation Module Framework Unit with a Bidirectional Gated Recurrent Unit (Bi-GRU) to enhance anomaly classification efficiency. Introducing a Structured Activation Loop Framework improves the model's ability to understand intricate details in short data analysis, while an Efficient Activation Module Unit reduces complexity and handles missing values in time series data. These advancements enhance how missing data is managed, while also integrating layered attention frameworks and advanced recommendation strategies, distinguishing this research from prior methods that often did not offer such complete solutions. In general, although previous research has made progress in area, the innovative features of the proposed prototypical represent a significant enhancement in utilizing (RNN) Recurrent Neural Networks for detecting network traffic anomalies without introducing new components.

Paper Organization

The current model conceptual framework is outlined as follows: section 1 gives an introduction of the model background. Section 2 examines existing research concerning anomaly prediction and problem detection. The next Section 3 specifically details the proposed methodology. Moreover, section 4 includes both a table and visual illustration of the data analysis. Section 5 examines the results of present study in comparison to previous research. In conclusion, section 6 offerings the findings of the study and offers recommendations for the future.

Literature Review

The section covers he traditional and advance methods by reviewing and explains about the examination in several conventional investigates of anomaly classification along with other methods for estimation on classification system in network traffic to ensure the security.

The conventional research has presented effective anomaly detection mechanism named as D-PACK. It comprises of supervised DL model like auto encoder and convolutional neural network (CNN) for auto profiling traffic patterns and extracting the abnormal traffic. It has used Mirai-CCU dataset [22]. Similarly, a variant DL architectures to the issue of anomaly prediction on network logs which are obtained by Sense firewall along with chief target the classification of event type in the prevailing system. It has utilised network intrusion dataset [23]. Correspondingly, a 5-layer auto encoder based model for a network anomaly classification tasks. This model has evaluated on NSL-KDD dataset that outperformed [24]. Congruently, CSE-CIC-IDS2018 dataset has been employed to evaluate the traditional model. It has applied CNN, RNN, LSTM, DNN, CNN+RNN and CNN+LSTM models have been constructed to identify the anomaly attacks in network and has attained better results [25].

In the same way, a CNN created technique has deployed for anomaly intrusion detection systems (IDS) which took IoT's advantage. It has provided qualities to analyse whole traffic across IoT. It has utilised NID and BoT-IoT dataset [26]. In parallel, a hybrid DL model has deployed to address various real-time and automatic surveillance techniques for abnormal detection to identify dynamic multitude activity in the applications of security. It has used Motion Emotion dataset [27]. Additionally, the existing model has deployed a DNN method for detection of anomaly in NID mechanism. It has utilised NSL_KDD dataset and soft max layer has been used along with cross entropy loss mechanism for enforcing the conventional model in various classification comprising 5 labels like normal, DoS, R2L, Probe and U2L. It has attained better results [28]. Likewise, a deep CNN model has been deployed for classifying and detecting nearest real time network intrusions from the imbalanced cloud surrounding. In addition to, the evaluation has been carried out with CSE-CIC-IDS2018 dataset [29].

Similarly, the classical model has presented a mechanism of anomaly detection utilising CNN and LSTM models. These models have been proficient on data network that are take out from the packet capture files. It has utilised a huge scale real network traffic dataset and benchmark intrusion detection dataset, CTU-13, ISCX-IDS and NSL-KDD datasets respectively [30]. In parallel, the existing system has evaluated several significant feature selection method for ML on DDoS detection. It has used NSL-KDD dataset which contains 41 features and 52,800 records. Has attained better accuracy using the features subset through the RFE method [31]. Contrastingly, conventional model has predicted various anomalies on various features in the dataset through applying ML models. It has utilised a dataset which is from Kaggle that comprises 357,952 samples along with 13 features with the lack of labelled datasets and noisy data[32]. Similarly the existed study namely split active learning anomaly detector (SALAD), KDD cup 199, and UNSW-NB 15 dataset using scikit-multiflow framework includes teaching auto encoders through real-time data to detect irregularities. Findings indicate SALAD is more effective than conventional methods in terms of both value of precision and speed [33]. Similarly, an approach existed namely Recurrent Extreme Learning based-boosted chimp (REL-BC) algorithm and extreme learning machine (ELM) used for anomaly detection, making it appropriate for cyber security [34] (Table 1).

PROBLEM IDENTIFICATION

Several existing researches have been limited by predicting the anomaly in the networks. It has several lacks and it is provided below.

- Limitations comprise the need for a substantial amount of high-quality training data and the risk of over fitting due to the auto encoder›s distinct training method, which may not adjust well to unfamiliar data distributions. Furthermore, challenges may arise in scaling and computational efficiency when dealing with extremely large datasets, as well as the need for accurate parameter tuning to improve performance in different scenarios [33, 34].
- Other limitations include the risk of over fitting with complex models, reliance on particular datasets that may not be relevant to all situations, and obstacles in implementing real-time processes due to the escalating computational demands of deep learning methods without incorporating new elements [29].
- Relying on particular training datasets restricts models' ability to adapt to new situations, as they frequently do not encompass all possible unusual behaviours. The complexities of the real world, like different lighting and crowd behaviour, make it harder to accurately detect things, and the computational requirements of deep learning slow down real-time processing.
- Furthermore, the process of obtaining and up keeping substantial amounts of labelled training data requires a lot of resources, resulting in high levels of false positives

Table 1. Deep learning–based anomaly and intrusion detection, highlighting datasets used, real-time performance advantages over traditional methods, and key limitations such as labeled data requirements and optimization challenges

Reference	Model name	Data sets	Methodology	Result	Limitations
[27]	CNN architecture	Motion Emotion Dataset (MED)	Employing optical flow and CNNs to capture the spatial-temporal features of crowd behavior.	The existed model has showed greater accuracy, reduced computational complexity, identifying anomalies	The reliance on labelled training data, which can be hard to collect fully. Fluctuations in lighting occlusion can influence detection performance.
[30]	CNN, LSTM	Benchmark intrusion detection dataset, CTU-13, ISCX-IDS and NSL-KDD datasets	The network flow where trained by source of data and files are captures from packed resources and assessed using benchmark intrusion detection datasets alongside a substantial real-world network traffic dataset.	These deep learning models greatly exceed conventional shallow learning techniques regarding anomaly detection efficiency, achieving real-time processing with minimal latency.	constraints, including the necessity for optimization strategies such as transfer learning, improve detection efficiency, difficulties in managing various data types
[22]	Variation Auto encoder (VAE), CNN architecture	Mirai-CCU dataset	To rebuild typical traffic patterns and detect anomalies through reconstruction errors	VAE identifies anomalies with high value of precision and low rates with false positives, surpassing conventional techniques in early detection abilities	Constraints such as the model›s vulnerability to the quality of the training data and possible difficulties in adjusting to fast-evolving network settings, which could influence its sustained efficacy in fluctuating situations.
[29]	Deep CNN architecture	CSE-CIC-IDS2018 dataset	This model CNN demonstrated to address class imbalance using methods like oversampling and cost-sensitive learning	Enhances detection accuracy and classification performance, attaining significant decreases in false positive rates, successfully identifies both frequent and unusual anomalies.	The possibility of over fitting owing to the model›s complexity and the requirement for significant computational resources, which could impede effective implementation in resource-limited settings.
[31]	Random Forest (RF) ML model	NSL-KDD dataset	The datasets used consist of synthetic traffic data created to reduce DDoS attacks and actual network traffic data.	Precisely detecting and alleviating DDoS attacks	Dependence on feature selection techniques, affect detection precision, computational burden linked to handling huge amounts of network data in real-time situations
[34]	(REL-BC) algorithm and (ELM)	NSL-KDD dataset	It effectively detect anomalies without relying on labelled data, making it appropriate for situations where anomalies are infrequent and varied.	Significant accuracy and resilience in identifying anomalies within diverse high-dimensional datasets,	Possible difficulties in generalizing across varied datasets, which could impact its efficiency in changing real-world scenarios.
[25]	CNN, RNN, LSTM, DNN, CNN+RNN and CNN+LSTM models	CSE-CIC-IDS2018 dataset	Mechanically learn features from raw network traffic data, facilitating effective anomaly detection.	Improved detection accuracy, false positive rate reduction.	Dependency on large amount of labelled data affects robustness.

that can be overwhelming for network administrators and lead to alert fatigue. The constantly changing network traffic requires constant adjustment to changing patterns, making it challenging for models trained on old data such as NSL-KDD to be effective.[23, 24] [27, 30].

Similarly, earlier approaches are followed to classify the network interruption by CNN, LSTM, REL-BC, and RF has been used to implement. It resulted in constraints like dependency on large amount of labelled data sets, difficulties in generalizing across varied dataset, possibility of over fitting, and vulnerability to the quality of the training data. While the approached method has overcome all of these difficulties by integrating the structured activation loop framework and efficient activation module unit with (UNSW-NB15) and NSL-KDD datasets. This approached method carries the overall performance and enhance the network anomaly classification by loop based frame work mechanism which has provided better accuracy and results.

MATERIALS AND METHODS

The research method classifies and extract data from the corresponding dataset for anomalies prediction. The prediction is approved by applying structured activation module framework unit with Bi-GRU, a deep learning model with activation function. Traditional works on the anomalies prediction have produced inaccurate results with slow speed convergence. Consequently, the respective model utilised the DL algorithm for prediction of UNSW-NB15 and NSL-KDD data. Furthermore, the flow of proposed DL model illustrated in below Figure 1.

Figure 1 deliberates the proposed flow, which indicates the proposed model creation on anomaly classification on network established environment, where it includes the loading the dataset, refers to the procedure of importing or accessing the dataset within the working environment. This probably involves importing data from sources such as CSV files, SQL databases, or various structured data formats. Also featuring data activities like dealing with missing values, outliers, transforming, normalizing, encoding categorical variables, therefore pre- processing is importance for enhancing the accuracy and resilience of models. Similarly, the separation is very much essential for assessing the model effectiveness and unfamiliar data, where this phase entails train the model using the refined training data and fine- tuning its parameters to improve performance. Finally an assessment procedure, Performance metrics are computed to assess the model exactness, value of precision, probability of detection, f1 measures that enhances classification. The following sections deliberates the functions used in present model and Figure 2 signifies the present model's architecture.

The Figure 2 represents the present model of architecture. It signifies that the used data are pre-processed and split as train and test set to classify the data to predict anomalies in the data. It may contains various types of attacks which has to be detected.

Dataset Description

The proposed system being suggested makes use of two publicly available datasets called UNSW-NB15 and NSL-KDD. However, both sets of data are commonly used in academic studies to create and evaluate machine learning methods focused on boosting the efficiency of IDS. NSL-KDD and UNSW-NB15 are both network datasets. That

Figure 1. Proposed flow mechanism.

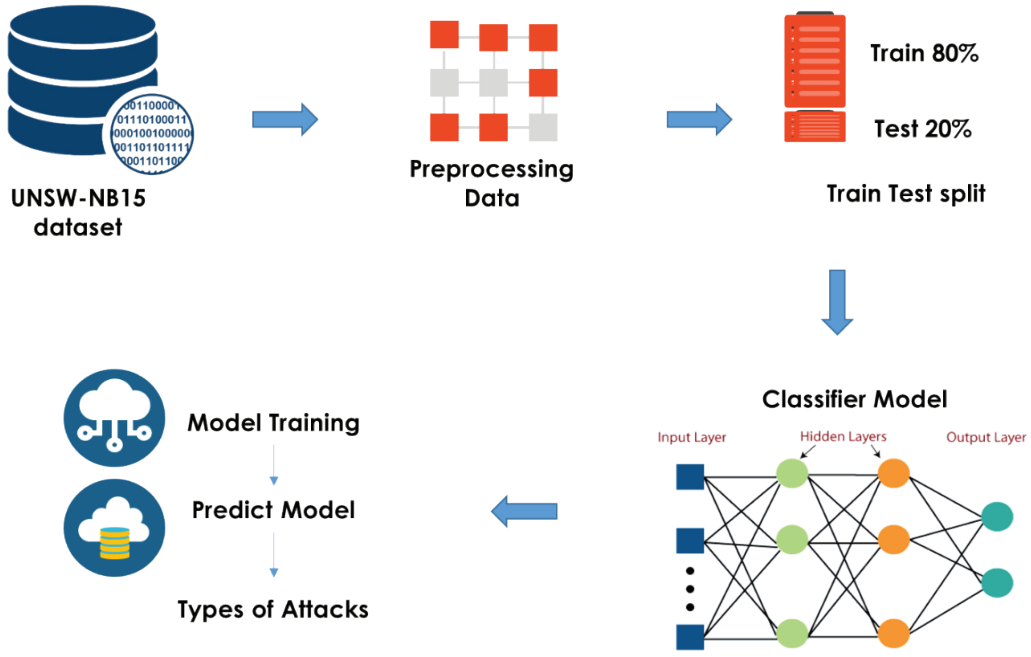

Figure 2. Present model architecture.

favoured by researchers because they offer thorough coverage of attacks, advanced features, extensive record sizes, authentic traffic simulation methods, and academic acknowledgment. On the other hand, iTrust and SWAT datasets are limited in their usefulness for research due to a lack of varied attack types and adequate data. Moreover, the variety of features and data range available make it a valuable resource for testing detection algorithms and enhancing machine learning effectiveness, which is why it is well-liked in academic circles. The data sets were obtained from the Kaggle platform.

UNSW-NB15 dataset

The UNSW-NB 15 dataset is a current dataset is used to estimating (IDS), it has comprised by a mixing of normal and malicious network traffic. This dataset consist of 9 kinds of attacks such as Denial of service (DoS), Fuzzers, Exploits, Generic, Shellcode, Analysis, Worms, Reconnaissance and Backdoors. It is a substantial and having 2, 57,673 record samples, where training set have 75,341 and testing set have 82,332 record samples. Each comprises with 48 features. It is a comprehensive majorly used for network intrusion detection mechanisms. The website line of UNSW-NB15 dataset is given below:

https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15/data

NSL-KDD dataset

Train set of NSL-KDD data contains labels of attack-type and levels in CSV format. It is N reviewed version of innovative KDD dataset, aimed at giving a more balanced and representative sample for intrusion detection research. Moreover, it consists total of 1, 48,517 samples of record that have 42 features in NSL-KDD dataset. Where attacks have been categorized likely DoS, User to Root (U2R), Remote to Local (R2L), probing, and normal connections. The records in test and train sets are reasonable that made reasonable to do the tests on entire set without any necessity for randomly choose a trivial portion. The official link of the NSL-KDD dataset is provided below:

https://www.kaggle.com/datasets/hassan06/nslkdd

Data Pre-Processing

The technological skill to modify raw data into proper data sets is called pre-processing, that pre-processes mavericks, missing values, noisy signals, feature scaling, label encoding, and other inconsistences before they are applied to the algorithm. In addition, the pre-processing stages the feature extraction and classification performance of the presented method. To achieve this, the designed system has incorporated two significant pre-processing methods such as missing values checking and feature scaling. While scaling features, the data features or variables range have been normalized to increase the effectiveness of the proposed model. The process is known as data normalization. Normalization is a data pre-processing technique employed to convert features in a dataset to a uniform scale, enhancing the effectiveness and precision of ML algorithms. The primary aim of normalization is to remove possible biases and distortions resulting from the varying scales of features. Here, Min-Max scalar normalization also known as Min-Max scaling, data has been used to convert into specific range, often (0, 1). It rescales to ensure the data

to map in the lowest value to 0, and scales the extreme value to 1(or -1). It is simple and easy to protect the relationships between data points, also suitable for algorithms data within bounded range assured. Similarly missing value handling is a biggest step in data pre-processing missing data, can evolve from various sources, such as human error, technical issues, or the inherent nature of the data collection process, these missing values can be handled by entire row list wise deletion by filling based on other available data through imputation techniques, advanced imputation includes interpolation, this is particularly used for time-series data where it include linear and polynomial interpolation , on the other hand multiple imputation also used for imputing missing values by multiple times. Similarly libraries and functions, arbitrary value replacement also utilized for missing value handling.

Data Splitting

DL devotes the data management to the aim of removing the overfitting of data. Essentially, DL employs data splitting as a tool to train the respective model whereby the training data is injected into the proposed method to enable the training stage parameters. Once the training has been accomplished, testing data are used to evaluate the deployed model for dealing with the test set. The current model splits the original data into two sets of 80:20, thus eighty percent of the new observations are used for training, while the remaining 20 percent of the observations are utilized for testing to calculate the performance of the respective technique.

CLASSIFICATION- STRUCTURED ACTIVATION MODULE FRAMEWORK UNIT WITH BI GRU

A Bi-GRU with update and reset gates is a variant of the standard GRU architecture, a RNN type which incorporates additional gates to control the flow of information through the network. The Bi-GRU with update and reset gates consists of two hidden states, one for the forward pass and one for the backward pass. The update gate controls the amount of data from the former hidden state that is carried out to the present hidden state. The reset gate controls the degree to which the preceding hidden state is reset before the new information is incorporated. As it described, RNN are designed to handle the sequence data. Moreover, RNN has its ability to learn the long term dependency. To solve these problems, some RNN structures are customised like GRU. Compared to other models, GRU utilises less parameters. Bi GRU is effective for tasks such as appropriate text representation, missing data value prediction, etc. and moreover, along with the preceding values, succeeding data also available. However, it comes with increased computational cost, over fitting potential, and difficulty in optimization. Hence the proposed model employs structured activation module framework unit with Bi GRU to work well in prediction.

The Bi-GRU model's gating mechanisms are improved by the Structured Activation Loop Framework, which introduces a structured method for controlling information flow more effectively according to input context. This system focuses on being able to adapt to different contexts by incorporating more contextual details into the activation

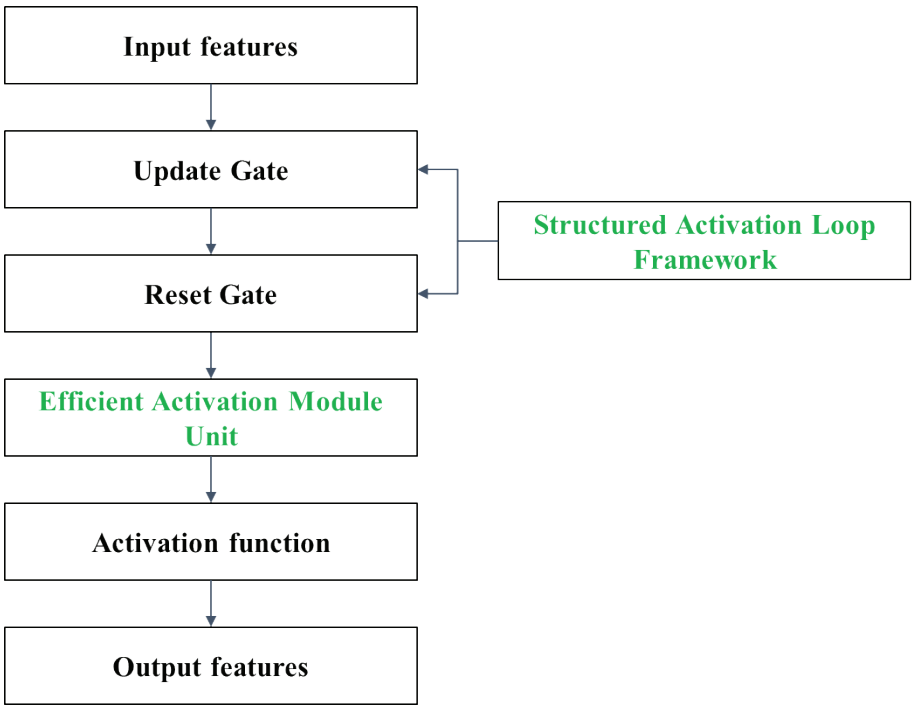

Figure 3. Mechanism of Present DL Model.

functions of the update and reset gates. This allows the model to accurately detect quick changes in emotions in brief reviews. Moreover, it enhances information recall, leading to improved retention of important details over time, which is crucial for identifying delicate hints in user attitudes expressed in reviews. Furthermore, the organized framework allows for flexible changes, allowing the model to adjust the impact of previous states reliant on present inputs, ultimately improving performance in tasks that involve understanding time sequences. In addition, the Efficient Activation Module Unit aims to improve the computational effectiveness of the Bi-GRU, especially in situations with sparse or missing data. This is accomplished by reducing the computational complexity of the model's activation functions, resulting in faster processing times, particularly when dealing with large datasets. The unit is equipped with methods for managing missing values, increasing resilience by enabling the network to focus on important data sections while ignoring unimportant ones. It helps enhance learning efficiency by enabling quicker convergence in training and reducing noise from data gaps, resulting in more dependable predictions and insights.

The Figure 3 represents mechanism of current DL model. Usually, a standard GRU approach includes a reset gate system and an update gate that helps in recognizing the input data. The update gate controls the data flow at every time step, whereas the reset gate eliminates the unrelated value, returning it to the update gate for further processing. This system determines the quantity of data retained from the past and the amount that is eliminated. Consequently, this method involves a decrease in prediction accuracy, extended outputs, and is time-consuming. In this suggested structured activated loop model, the error data is observed within the update gate without going through the reset gate; it can be confirmed several times in a loop format. This mechanism captures the behaviour of short reviews, guarantees the efficient retention of relevant information over time, adjusts historical states based on present input, and involves a lengthy procedure. In the effective activation module unit, class data is divided efficiently to precisely predict the appropriate output required for handling missing values. This allows the network to concentrate on important segments while excluding irrelevant or uninformative components of the input, which improves resilience, reduces noise, and leads to more dependable predictions and insights.

Since Bi-GRUs are a potent tool for anomaly classification, however it also have several disadvantages. These disadvantages should be considered when choosing a model for anomaly classification. To overcome the issues, activation function is incorporated in Bi-GRU model. The optimal of activation function considerably impact the capability for network to learn multifaceted patterns, generalize to unseen data, and differentiate among usual and abnormal data. After process of gate function, the data were passed to efficient activation module unit which is used to develop the accuracy of the present model. Moreover, activation function segregates the class data precisely to predict the appropriate output features. In addition, Bi-GRU uses gating mechanism to embrace the memory without separate unit. Both of the gates control and update data in time t. At the time t, it updates the data as follows.

$$o_t = (1 - ug_t) \odot o_{t-1} + ug_t \odot \widetilde{o_{t-1}} \quad (1)$$

o_t – Represents the output time t, o_{t-1} – previous time step t-1, ug_t – Term controls the signals, weight, or input forms at time t, ⊙ - Indicates an element- wise multiplication, which allows for selective scaling of features based on their relevance. The above equation signifies as linear function to integrate previous state o_{t-1} and present state $\widetilde{o_{t-1}}$, which is measured by new sequence data. The traditional GRU measure ug_t in following equation (2).

$$ug_t = \sigma\,(w_{ug} y_t + U_{ug} o_{t-1} + m_{ug}) \quad (2)$$

ug_t – Represents the amount produced or state variable at time t, y_t – denotes an input or state variable at time t, o_{t-1} – Previous state or time from an input t-1, m_{ug} – indicates a constant parameter influence the output, w_{ug} – A weight or coefficient applied to the variable y_t, σ – Initiates initial transformation for the context. This equation (2) represented for processing sequential data such as time- series. Here, y_t current input sequence. Bi GRU process the input sequence as $\hat{y} = y_{t-1}, y_t, y_{t+1}$ to replace with y_t to fetch more information. And the calculation of structured activation module unit with Bi-GRU calculates as follows

$$ug_t = \sigma\,(w_{ug}\hat{y} + re_t \odot (U_h o_{t-1}) + m_o \quad (3)$$

w_{ug} – A weight parameter linked with the input $\hat{y}$. It scales the contribution of to the overall output ug_t, m_o –similarly a bias term added to the equation, providing an offset that can help adjust the output independently of the inputs, $\hat{y}$ – Usually denotes various predicted output or feature vector from past calculations or network layers. The reset gate re_t modulates how much of the previous state contributes to the calculation. Equation (3) makes suitable for applications where outputs need to be interpreted probably in recurrent neural networks.

The temporary state $\widetilde{o_t}$ can be calculated through the equation (4)

$$\widetilde{o_{t-1}} = \tanh\,(w_{ug}\hat{y} + re_t \odot (U_o o_{t-1}) + m_o \quad (4)$$

ret⊙ (Uoot−1) integrates information both present and previous states, permitting for dynamic modifications. The term $w_{ug}\hat{y}$ contributes to the modified state based on feature prediction, the bias term m_o helps to tune the output finely before applied into the activation function. The *tanh* function ensures that $\widetilde{o_{t-1}}$ remains within a bounded range, making it suitable for applications where output needed to be interpreted or where non- linear transformations are beneficial. This equation calculates a temporary state based

on both current and previous states, transformed by a hyperbolic tangent function to ensure values are between -1 and 1. Where y_t is replaced by $\hat{y}$. The reset gate re_t is weight considering that how much the model keep the previous state. If re_t = 1, then the state need to keep whole previous state o_{t-1}. In modified Bi-GRU, reset gate is calculates as follows

$$re_t = \sigma\left(w_{re}\hat{y} + (U_{re}o_{t-1}) + m_o\right) \quad (5)$$

After the gate mechanism, assuming a review contains l_s reviews where it have contains l_w reviews. Here, w_{it} refers to *t th* one and $t \in [1, L_w]$, $i \in [1, L_s]$, it also employs an embedding matrix w_e to plot w_{it} into vector y_{it}. A Bi-GRU a bidirectional unit, it has forward direction $\overrightarrow{weight\ input}$ and reverse direction.

$$y_{it} = w_e w_{it}, t \in [1, L_w] \quad (6)$$

Where equation (6) maps vector representation using an embedding matrix. The w_e acts as a scaling factor for the variable w_{it}. Where the resulted product gives the value of y_{it} that might represent some form of accumulated information, prediction, or transformation based on the input weights and parameters.

$$\vec{o}_{it} = \overrightarrow{weight\ input}\,(y_{it}), t \in [1, L_w] \quad (7)$$

$$\overleftarrow{o}_{it} = \overrightarrow{weight\ input}\,(y_{it}), t \in [L_w, 1] \quad (8)$$

The, it combines $\vec{o}_{it}$ and $\overleftarrow{o}_{it}$ as $[\overrightarrow{o}_{it}:\overleftarrow{o}_{it}]$ that contains all data taking as o_{it} as center. Hence, the mechanism employs attention mechanism to compute various weights for each anomaly. It use the following function to measure the attention weights, while these equations (7) & (8) compute forward and backward states for each time step in a bi directional manner.

$$c_{it}^w = \sigma(W_w\, o_{it} + m_w) \quad (9)$$

$$a_{it}^w = \frac{exp\,(c_{it}^w)^T c^w}{\sum_t exp\,(c_{it}^w)^T c^w} \quad (10)$$

These equations (9) & (10) calculate attention weights based on the context vector for each evaluation, allowing the model to focus on applicable parts of the input data and also it acts an attention mechanism for anomalies.

$$s_i = \sum_t a_{it}^w\, m_{it} \quad (11)$$

s_i- An aggregated measure, or an output from a model for the specific index, ∑t- summing over the variable t. this implies that the equation considers multiple time steps or instances, contributing to the overall value of s_i, a_{it}^w – Weight or coefficient liked with an index *i* at time *t*, raised to the power of *w*. The exponentiation suggests that the weights may have a non- linear influence on the sum, which is important in contexts to emphasize or de- emphasize different contributions, m_{it} – Another variable associated with index *i* at time *t*. it could represent a measurement, feature, or any relevant quantity that is being multiplied by the weighted term a_{it}^w.This equation (11) computes a value s_i by combining contributions from multiple instances. The summation denotes that all these contributions are joined to give a single output for each index *i*. Moreover, it needs to compute different data weights to affect the sentiment reviews, where an s-attention applies as w-attention

$$c_i^s = \sigma(W_s\, m_i + o_s) \quad (12)$$

$$a_i^s = \frac{exp\,(c_i^s)^T c^s}{\sum_t exp\,(c_i^s)^T c^s} \quad (13)$$

The above (12) & (13) equations were the secondary attention mechanism defines next attention mechanism focused on different datasets to contribute overall consideration. It signifies a score, total or any accurate calculations on dependency.

$$v = \sum_i a_i^w\, m_i \quad (14)$$

v – Represents the final computed value resulting from the summation, a_i^w – Variable raised to the power of *w*. The exponential illustrates every a_i is converted into non-linear before multiplied by m_i and m_i could represent a measurement, feature, or any relevant quantity that is being multiplied. This equation (14) contribution aggregates from all reviews into a single vector illustration from multiple instances. The summation implies that all these contributions are combined to produce a single output for *v*. Here, *v* is review level vector that consists almost all data and then to build loss function deliberated as follows.

$$Loss = \sum_{r \in R}(v - mt)^2 \quad (15)$$

Similarly, equation (15) represents the loss function for training process, it measures far predictions from actual values by summing squared difference across all analysis. Finally the equations collectively describe a complex neural network architecture that efficiently processes sequential data through gated mechanisms and attention mechanisms. The integration of bidirectional processing allows for capturing contextual information from both past and future states, making this model particularly powerful for tasks like sentiment analysis, anomaly detection, or any other applications involving sequential data.

RESULTS AND DISCUSSION

Performance Metrics

Performance metrics basically serve as a means of gauging the efficiency of the intended research through the use of various metrics such as probability of detection rate, value of precision, exactness, and F1 measures value.

Recall Metric

In this case, recall serves to research the proportion of data that was correctly identified by the corresponding model. The recall formula is given by the following equation (16),

$$\text{Recall} = \frac{\text{True\_Pos}}{\text{False\_Neg+True\_Pos}} \quad (16)$$

Accuracy Metric

Exactness is the main metric that is used to analyse the number of estimates which are almost correct in the present model. The exactness formula is shown in equation (17),

$$\text{Accu} = \frac{\text{TP+TN}}{\text{TP+FP+TN+FN}} \quad (17)$$

TP, FP, TN, FN are True Positive, False Positive, True Negative, and False Negative, respectively.

F1 Score Metric

F1-measures is aimed at testing the correct predictions of the positive class in the current model. The f1-score formula is referred to in equation (18),

$$\text{F1 Score} = 2 * \frac{\text{Recall*Precision}}{\text{Recall+Precision}} \quad (18)$$

Precision Metric

One of the precision metrics is known as the method's covariance unit, which is achieved by suitably predicted cases (*True_Pos*) to the total number of cases that have been precisely considered (*True_Pos* + *False_Pos*). Equation (19), shows the formula for precision.

$$\text{Precision} = \frac{\text{True\_Pos}}{\text{False\_Pos+True\_Pos}} \quad (19)$$

Performance Analysis

Various evaluation metrics such as precision, recall, F1-score and accuracy are used to test the effectiveness of the current DL model. Similarly, Confusion Matrix (CM) is used for understanding the performance of the proposed research. It contains and predicts the performance of the classification algorithm. Hence, the CM shows the number of correct as well as incorrect predictions made by the class. The Figure 4 depicts the CM for the proposed NSL-KDD dataset.

The Figure 4 signifies the CM of NSL-KDD Dataset. It is used to depict multi-class classification of prediction results which contains the prediction results summary outcomes of all instances of utilised dataset for the testing process. The matrix contains 6 classes with values up to 18,380, indicating a subset or broader categories for analysis. High diagonal values show correct predictions, especially for class 0 and 1. Classes 2-5 have fewer correct predictions but still perform well. Misclassifications are sparse, with some confusion between class 0 and 5, and 1 and 5. Other misclassifications are minimal, such as class 4 as class 0. Compared to the first matrix, misclassifications are reduced due to fewer classes, this model shows strong performance

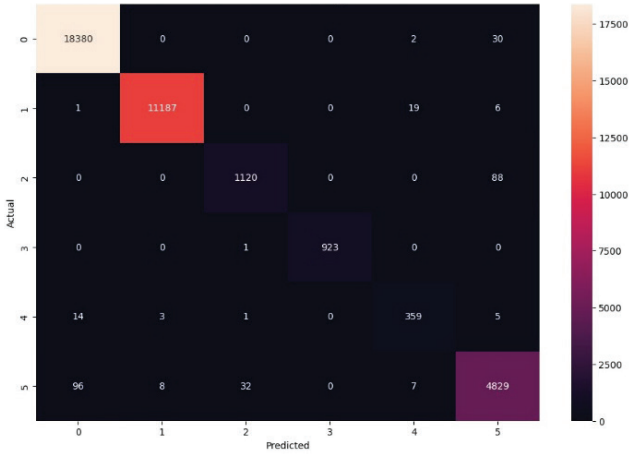

Figure 4. CM for NSL-KDD dataset.

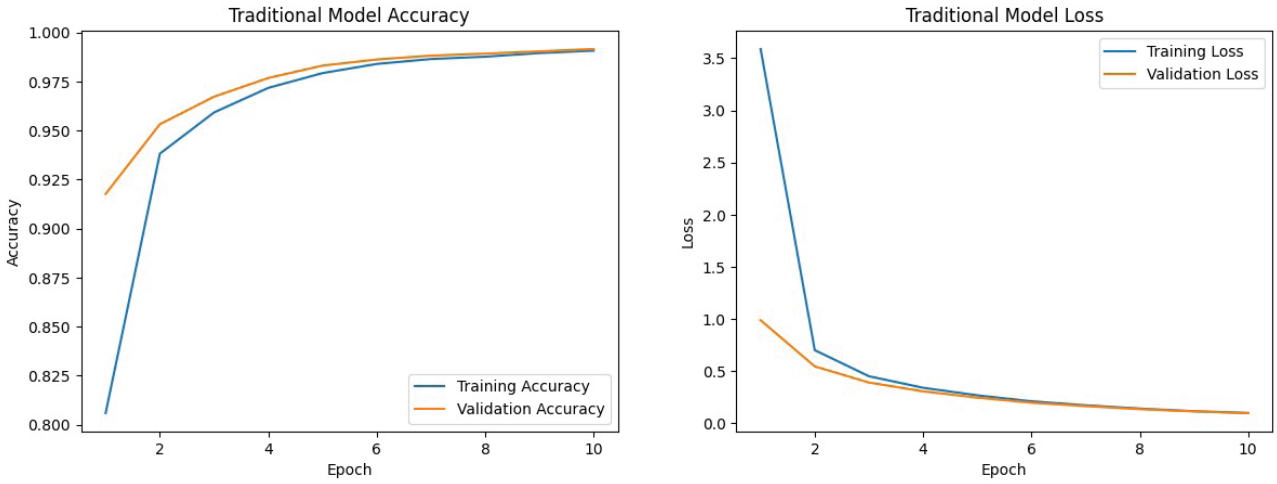

Figure 5. Model accuracy and loss of NSL-KDD data

with lower confusion. Additionally, Figure 5 represents the model exactness and model loss for NSL-KDD Data.

The Figure 5 denotes the model accuracy and loss of NSL-KDD dataset. The exactness of model gets increased in increasing epoch's counts in both training and validation set whereas the model loss is maintained low up to the 10 epochs in validation. The model loss in training is low in all epochs count.

The first graph model accuracy shows the accuracy changes with epochs during the model training. In the beginning epochs (1-3) both training and validation accuracy improve quickly from around 0.82. Middle epochs (4-6) see high accuracy levels around 0.97-0.98, with validation slightly better. Later epochs (7-10) stabilize around 0.98, showing near optimal performance.

The second loss model graph reveals key insights into its training process. Initially high, both training and validation losses quickly decline in the first few epochs. Around 3-4 epochs, loss reduction slows, signalling convergence by epoch 10. Despite lower training loss, better generalization is observed.

The Table 2 illustrates the classification result for NSL-KDD dataset. It shows the value of precision, exactness, recall and f1-measures of the present model. All models have achieved a 99% accuracy rate. However other Classes shows a precision, recall, F1- scores of 0. 93, 0 94, 0. 97, 0.95. Classes 0, 1, and 3 showed perfect recall of 1 and precision scores on class 1 and 3. The Table 3 signifies the proposed NSL-KDD dataset evaluation.

The Table 3 denotes the metrics of proposed model with NSL-KDD dataset. It deliberates that the present model attains 0.99, 0.97, 0.97, 0.97 of exactness, value of precision, probability of detection and f1-measures respectively and giving enhanced results. Considerably, Figure 6 signifies the model exactness and model loss of UNSW-NB15 dataset.

The Figure 6 denotes the model exactness and loss of UNSW-NB15 data. The model accuracy gets improved in increasing epoch's counts in training set and validation set whereas the model loss is maintained low up to the 10 epochs in validation. The model loss in training is low in all epochs count. Moreover, Figure 7 depicts the CM of NSL-KDD dataset.

Traditional model accuracy graph depicts accuracy value from 0.9575 to 0.9775, showing the model performances over epochs. When accuracy quickly arises, while minor fluctuations occur later on, with validation accuracy dipping slightly between epochs 6 and 8 before stabilizing around 0.975. This shows the model is not over fitting.

Table 2. Classification result on NSL-KDD dataset

Classes	Accuracy	Precision	Recall	F1-score
0	0.99	0.99	1	1
1	0.99	1	1	1
2	0.99	0.97	0.93	0.95
3	0.99	1	1	1
4	0.99	0.93	0.94	0.93
5	0.99	0.97	0.97	0.97

Table 3. Metrics of NSL-KDD Dataset

Model	Exactness	Value of precision	Recall	F1-score
Proposed method	0.99	0.97	0.97	0.97

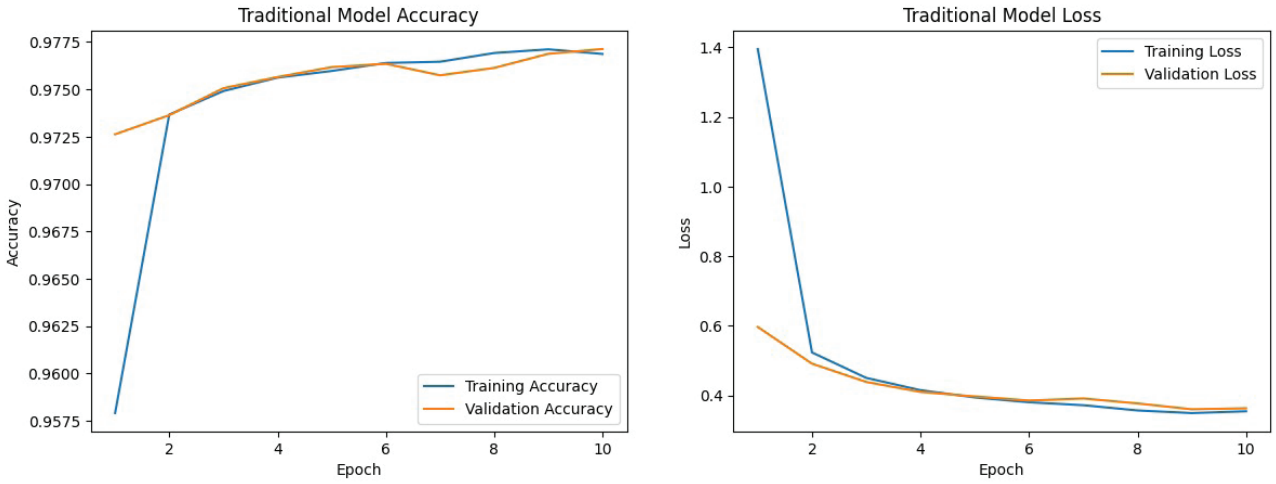

Figure 6. Model accuracy and loss for UNSW-NB15.

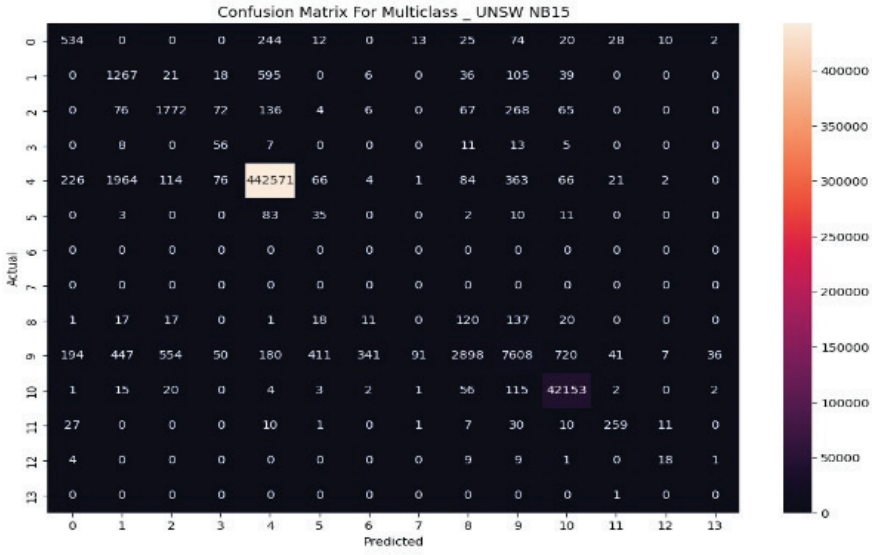

Figure 7. CM of UNSW-NB15 data.

The loss model graph provides insights into the model training process. In the early epochs, the training loss rapidly decreases from above 1.4 to below 0.5, indicating learning of essential data patterns. Middle epochs shows stable losses around 0.4, with training slightly lower than validation. Later epochs shows both losses stabilize below 0.4, indicating well trained convergence.

The Figure 7 signifies the CM of NSL-KDD Data. It is used to depict multi-class classification of prediction results which contains the results summary outcomes of all instances of utilised UNSW-NB15 dataset for the testing process. It contains actual and predicted labels. The matrix reveals imbalanced data with values exceeding 400,000. High diagonal values show accurate predictions, notably in cell (4, 4) at 442,571 for "normal" network traffic. Class 9 and 11 also have high true positives but lower than class 4. Misclassifications exist in off- diagonal cells, with class 9 having the most. Furthermore, Table 4 depicts the classification result for UNSW-NB15 dataset.

Information in Table 4 presents various classification metrics for the UNSW-NB15 dataset and describes the state of art model. What is more, the figure presents a performance evaluation in terms of precision, recall, f1-score and accuracy for the model under discussion. So, all the classes could classify correctly at 98% of the cases. Class 10 got the highest result of accuracy, precision, recall and f1-score. In

Table 4. Classification result on UNSW-NB15

Classes	Exactness	Value of Precision	Probability of Detection	F1-measures
0	0.98	0.54	0.56	0.55
1	0.98	0.33	0.61	0.43
2	0.98	0.71	0.72	0.71
3	0.98	0.21	0.56	0.3
4	0.98	0.32	0.99	1
5	0.98	0.06	0.24	0.1
6	0.98	0.25	0.32	0.36
7	0.98	0.32	0.25	0.69
8	0.98	0.04	0.35	0.07
9	0.98	0.87	0.56	0.68
10	0.98	0.98	0.99	0.99
11	0.98	0.74	0.73	0.73
12	0.98	0.38	0.43	0.4
13	0.98	0.21	0.69	0.38

Table 5. Metrics of UNSW-NB15 dataset

Method	Precision	F1-Score	Accuracy	Recall
Proposed method	0.99	0.98	0.98	0.98

Table 6. Comparative performance of NSL-KDD dataset with existing models

Model	Precision	Accuracy	Recall	F1-Score
DNN-1 Layer	0.82	0.91	0.92	0.92
DNN-2 Layer	0.81	0.91	0.91	0.91
DNN-3 Layer	0.83	0.92	0.78	0.85
DNN-4 Layer	0.91	0.94	0.72	0.83
Proposed model	**0.97**	**0.98**	**0.97**	**0.97**

like manner, classes 1 and 3 were at the moderate levels of recall rates. Class 4 had an ultimate f1-score equal to 100%, while class 5 was f1- scored 1. Moreover, information in the Table 5 points to the proposed model metric results in the UNSW-NB15 dataset.

The Table 5 indicates the evaluated metrics of proposed model in UNSW-NB15 dataset. It deliberates that the proposed model attains 98%, 99%, 98% and 98% of exactness, value of precision, probability of detection and f1-measures respectively.

Comparative Analysis

The section exemplifies the comparative analysis of the present model depending on various dataset used in the respective model. The Table 6 illustrates the performance comparative of NSL-KDD dataset with various conventional researches.

The Table 5 represents qualified performance of NSL-KDD dataset with existing models. The present model attains 0.07, 0.07, 0.06 and 0.04 of accuracies more than DNN-1 Layer, DNN-2 Layer, and DNN-3 Layer and DNN-4 layer respectively [35]. Moreover, signifies the performance metrics comparison of existing model. Deliberates the metrics comparison for NLS-KDD dataset. It shows that the proposed research model attains higher values in all the metrics evaluated and shows the improved efficiency than other existing models. Furthermore, the Table 7 indicates the comparison of accuracies with conventional model for NSL-KDD dataset.

The Table 7 depicts the comparison of accuracies with existing models with proposed model. It demonstrates that the proposed research attained 0.98 of accuracy in NSL-KDD dataset whereas, 0.81, 0.74, 0.73 and 0.71 of accuracies are attained by DNN, RNN, DBN and LSTM respectively [36]. Additionally, Table 7 shows the accuracy comparison with conventional research models. It clears that the proposed model achieves better accuracy metric results than other prevailing models such as DNN, RNN, DBN and LSTM models respectively. Similarly, the Table 8 shows the comparative performance of UNSW-NB15 dataset with prevailing models.

The Table 7 exemplifies the comparison performance of UNSW-NB15 dataset with conventional models. It shows that the respective model attains better results than other prevailing models [37]. It attained 0.98, 0.98, 0.98 and 0.98 of exactness, probability of detection, F1-measures and value of precision respectively whereas other models such as LR, SVM, DT, Auto-encoder, BAE0, BAE 1, BAE 2 and BAE have attained less than 0.91 of accuracy. It shows the efficiency of the present model than other such models. The Comparison Analysis on Existing Models for

Table 7. Comparison of accuracies of NSL-KDD dataset

Model	Accuracy
DNN	0.81
RNN	0.74
DBN	0.73
LSTM	0.71
Proposed model	**0.98**

Table 8. Comparative performance of UNSW-NB15 dataset with existing models

Methods	Exactness	Probability of detection	Value of Precision	F1-Measures
LR	0.8951	0.8	0.62	0.7004
SVM	0.9075	0.7011	0.6949	0.698
Decision tree	0.897	0.8541	0.6173	0.7166
Auto-encoder	0.822	1	0.4566	0.6269
BAE 0	0.886	1	0.4063	0.5778
BAE 1	0.90002	1	0.8198	0.901
BAE 2	0.8425	0.8229	1	0.955
BAE	0.8761	0.8649	0.742	0.8113
Proposed method	**0.98**	**0.98**	**0.99**	**0.98**

Table 9. Accuracy comparison with prevailing models

Methods	Exactness
GA-DBN	0.8599
DBN	0.82
CDBN	0.8229
KG-CRBN	0.8649
Proposed method	0.97

Table 10. Comparative analysis with existing models for UNSW-NB15 dataset

Methods	Precision	Accuracy
SVM	0.6949	0.9075
AUTO ENCODER	0.4566	0.822
Proposed Method	0.99	0.98

respective dataset. It shows that the present model attained better metric results than existing models. The proposed model attained 0.0849, 0.0725, 0.083, 0.158, 0.188, 0.07998, 0.1375 and 0.1039 of accuracies more than LR, SVM, DT, Auto-encoder, BAE0, BAE 1, BAE 2 and BAE respectively. Moreover, Table 9 describes the accuracy comparison with conventional models.

Table 9, deliberates the accuracy comparison for UNSW-NB15 dataset with prevailing models. It shows that the proposed model attains 0.97 of accuracy [38]. Other conventional models such as GA-DBN, DBN, CDBN and KG-CRBM models have attained 0.8599, 0.82, 0.8229 and 0.8649 of accuracy respectively. The comparative analysis on accuracy for UNSW-NB15 dataset. It denotes that the proposed model achieves better accuracy metric results than other prevailing models such as GA-DBN, DBN, CDBN, KG-CRBM models respectively. It attained 0.1101, 0.15, 0.1471 and 0.1051 of accuracy more than GA-DBN, DBN, CDBN, and KG-CRBM respectively.

The Table 10 depicts the comparison of proposed work with the existing models [37]. Here the methods SVM, auto encoder has delivered 0.6949, 0.4566 and 0.9075, 0.822 of precision metrics and accuracy, while the proposed has exhibited high range of 0.99 and 0.98 in precision accuracy performances.

As organizations grow more reliant on interconnected networks, it becomes crucial to anticipate deviations in unusual network activities, which is vital for ensuring security. Unusual behaviours or developments in network security are significantly rising in the real-time environment. Thus, anomaly classification focuses on recognizing unusual patterns or behaviours. The proposed model has been improved with a structured activation loop framework integrated with a bidirectional gated recurrent unit. In this setup, the pre-processed training data is separated into training and testing processes. Existing datasets were evaluated with various Deep Neural Network (DNN) layers 1, 2, 3, and 4, obtaining precision, accuracy, recall, and F1 scores of 0.82, 0.83, 0.91, 0.92, 0.95, and 0.78, respectively. However, the pre-processed data from the proposed model has demonstrated superior performance scores of 0.97, 0.98, 0.97, and 0.97 on NSL-KDD datasets. Likewise, the UNSW-NB15 dataset has recorded scores of 0.98, 0.98, 0.99, and 0.98 in the metrics of exactness, value of precision, probability of detection and F1-measures when associated to earlier models such as LR, SVM, decision tree, auto encoder, BAE, BAE0, 1, and 2, which have lower scores of 0.62, 0.6949, 0.7004, 0.8761, 0.4566, 0.4063, etc., across all metrics. Nonetheless, traditional models such as DBN, CDBN have achieved accuracies of 0.8599, 0.82, 0.8229, and 0.8649. It indicated that the proposed model has achieved higher accuracies of 0.97. Consequently, the outcomes from the testing and training efforts produce improved results and enhance the ability to detect emerging anomalies in network and cyber security against hackers and malicious threats.

Applications of Activation Mechanism in Network Security

The proposed work has emerged for protecting and classifying anomalies with an activation mechanism for anomaly classification has numerous key applications in network cyber security includes the data security in protecting confidential information from unauthorized access and creating occurrence response measures to promptly address security incidents. By employing data analysis to detect possible threats through patterns and behaviours, enabling organizations to proactively tackle vulnerabilities. Also implementing Intrusion Prevention Systems to identify and prevent harmful actions in real-time, safeguarding the network against diverse cyber threats. Similarly, splitting the network- connected devices such as laptops and mobile devices against threats using antivirus programs and device management policies.

CONCLUSION

Anomaly classification was a crucial aspect of network security that helped organizations detect and address potential threats and security in real-time. Through the identification and classification of anomalies, security teams were able to quickly identify and mitigate possible security threats, consequently lowering the chances of a successful cyber-attack. Therefore, efficient classification of anomalies was essential for ensuring network security and data storage. However, the precision of inspections by human experts was slow and constrained. To address this issue, the proposed study of the Structured Activation Module Framework Unit monitored error data directly in the update gate without requiring the reset gate, allowing for numerous checks in a loop format. The activation module unit effectively divided the class data to forecast relevant output characteristics by resolving missing values.

The study employed the NSL-KDD and UNSW-NB15 datasets as a demonstration for the effectiveness of the present model. With the NSL-KDD dataset, the study recorded its metrics of accuracy, precision, recall, and F1 score at 0.99, 0.97, 0.97, and 0.97 correspondingly. Whereas for the UNSW-NB15 dataset, the model's performance was measured in terms of accuracy, precision, recall, and F1-score, which were 0.98, 0.99, 0.98, and 0.98 respectively. Hence, the proposed research overcame challenges related to scaling, over fitting issues, and handling large datasets with inaccurate patterns and networks. Nonetheless, it also recognized specific limitations within the datasets, including a lack of representation for certain attack classes and persistent issues concerning feature relevance and applicability of results to real-life situations. In the future, the techniques established in this study showed potential for use across different network intrusion datasets, improving predictive abilities and ultimately sustaining cyber security strategies against unauthorized access and evolving cyber threats. The proposed study performed superiorly in both NSL-KDD and UNSW-NB15 datasets; however, this method would be considered for application in other datasets in future research endeavours.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that supports the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ETHICS

There are no ethical issues with the publication of this manuscript.

STATEMENT ON THE USE OF ARTIFICIAL INTELLIGENCE

Artificial intelligence was not used in the preparation of the article.

REFERENCES

[1] Inuwa MM, Das RJIT. A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks. Internet of Things 2024;26:101162. [CroosRef]

[2] Khan S, Ali ZJMCJ. Neural Networks in Particle Detector Quality Assurance: A Deep Learning Approach. 2024;5:1–10.

[3] Ullah I, Mahmoud QIJHA. Design and development of a deep learning-based model for anomaly detection in IoT networks. IEEE Xplore 9;103906–103926. [CrossRef]

[4] Madasamy NS. DKRF: Machine Learning with Optimised Feature Selection for Intrusion Detection. 2023;57.

[5] Merizio IF, Chavarette FR, Moro TC, Outa R, Mishra VN. Machine learning applied in the detection of faults in pipes by acoustic means. J Inst Eng (India) Ser C 2021;102:975–980. [CrossRef]

[6] Hussein A, Chadad L, Adalian N, Chehab A, Elhajj IH, Kayssi AJJCST. Software-Defined Networking (SDN): the security review. J Cyber Secur Technol 2020;4:1–66. [CrossRef]

[7] Ahmad Z, Khan AS, Nisar K, Haider I, Hassan R, Haque MR, et al. Anomaly detection using deep neural network for IoT architecture. Apll Sci 2021;11:7050. [CrossRef]

[8] Hooshmand MK, Huchaiah MDJDTR. Network Intrusion Detection with 1D Convolutional Neural Networks. Scientific 2022;1:66–75. [CrossRef]

[9] Kumar D, Joshi AB, Singh S, Mishra VN. Digital color-image encryption scheme based on elliptic curve cryptography ElGamal encryption and 3D Lorenz map. AIP Conference Proceedings 2021;2364. [CrossRef]

[10] Hooshmand MK, Hosahalli DJTTIT. Network anomaly detection using deep learning techniques. Inst Eng Technol 2022;7:228–243. [CrossRef]

[11] Aversano L, Bernardi ML, Cimitile M, Pecori R, Veltri LJWC, Computing M. Effective anomaly detection using deep learning in IoT systems. Wirel Commun Mob 2021;1–14. [CrossRef]

[12] Xu Z, Wang Z, Xu J, Shi H, Zhao H. Enhancing Log Anomaly Detection with Semantic Embedding and Integrated Neural Network Innovations. Computers, Materials & Continua, 2024;80:3991–4015. [CrossRef]

[13] Outa R, Chavarette FR, Goncalves AC, Silva SL, Mishra VN, Panosso AR, et al. Reliability analysis using experimental statistical methods and AIS: application in continuous flow tubes of gaseous medium. Technology 2021;43:e55825. [CrossRef]

[14] Xie X, Wang B, Wan T, Tang WJIA. Multivariate abnormal detection for industrial control systems using 1D CNN and GRU. IEEE Xplore 2020;8:88348–88359. [CrossRef]

[15] Bakhshi T, Ghita BJS, Networks C. Anomaly detection in encrypted internet traffic using hybrid deep learning. Secur Commun Netw 2021;2021:5363750. [CrossRef]

[16] Rajasekhar RNV, Sreedivya N, Jagadesh BN, Gandikota R, Lella KK, Pydala B, et al. Enhancing

anomaly detection: A comprehensive approach with MTBO feature selection and TVETBOOptimized Quad-LSTM classification. Comput Electr Eng 2024;119:109536. [CrossRef]

[17] Gaffar A, Joshi AB, Singh S, Mishra VN, Rosales HG, Zhou L, et al. A Technique for Securing Multiple Digital Images Based on 2D Linear Congruential Generator, Silver Ratio, and Galois Field. IEEE Access 2021;9:96125–96150. [CrossRef]

[18] Rakha MA, Khan IU, Ouaissa M, Ouaissa M, Ayub MY. Hybrid Model for IoT-Enabled Intelligent Towns Using the MQTT-IoT-IDS2020 Dataset. Boca Raton: CRC Press; 2024. p. 159–176. [CrossRef]

[19] Kumar D, Joshi AB, Mishra VN. Optical and digital double color-image encryption algorithm using 3D chaotic map and 2D-multiple parameter fractional discrete cosine transform. Results in Optics. 2020;1:100031. [CrossRef]

[20] Gaffar A, Joshi AB, Kumar D, Mishra VN. Image encryption using nonlinear feedback shift register and modified RC4A algorithm. J Appl Mat Inf 2021;39:859–882. [CrossRef]

[21] Outa R, Chavarette FR, Mishra VN, Goncalves AC, Garcia A, Pinto SS, et al. Analysis and prognosis of failures in intelligent hybrid systems using bioengineering: Gear coupling. J Eng Exact Sci 2022;8:13673-01-18e. [CrossRef]

[22] Hwang RH, Peng MC, Huang CW, Lin PC, Nguyen VLJIA. An unsupervised deep learning model for early network traffic anomaly detection. IEEE Xplore 2020;8:30387–30399. [CrossRef]

[23] Fotiadou K, Velivassaki TH, Voulkidis A, Skias D, Tsekeridou S, Zahariadis TJI. Network traffic anomaly detection via deep learning. Information 2021;12:215. [CrossRef]

[24] Xu W, Jang-Jaccard J, Singh A, Wei Y, Sabrina FJIA. Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset. IEEE Xplore 2021;9:140136–140146. [CrossRef]

[25] Wang YC, Houng YC, Chen HX, Tseng SMJS. Network anomaly intrusion detection based on deep learning approach. Sensors 2023;23:2171. [CrossRef]

[26] Saba T, Rehman A, Sadad T, Kolivand H, Bahaj SAJC. Anomaly-based intrusion detection system for IoT networks through deep learning model. Comput Electr Eng 2022;99:107810. [CrossRef]

[27] Rezaee K, Rezakhani SM, Khosravi MR, Moghimi MKJP. A survey on deep learning-based real-time crowd anomaly detection for secure distributed video surveillance. Springer Nature Link 2024;28:135–151. [CrossRef]

[28] Hussien ZJBSJ. Anomaly detection approach based on deep neural network and dropout. Baghdad Sci J 2020;17:701. [CrossRef]

[29] Vibhute AD, Nakum VJPCS. Deep learning-based network anomaly detection and classification in an imbalanced cloud environment. Procedia Comput Sci 2024;232:1636–1645. [CrossRef]

[30] Arjunan T. Real-time detection of network traffic anomalies in big data environments using deep learning models. Int J Res Appl Sci Eng Technol 2024;16:1–11.

[31] Nadeem MW, Goh HG, Ponnusamy V, Aun YJC. DDoS detection in SDN using machine learning techniques. Comput Mater Contin 2022;71.

[32] Mukherjee I, Sahu NK, Sahana SK. Simulation and modeling for anomaly detection in IoT network using machine learning. Int J Wirel Inf Net 2023;30:173–189. [CrossRef]

[33] Nixon C, Sedky M, Champion J, Hassan M. SALAD: A split active learning based unsupervised network data stream anomaly detection method using autoencoders. Expert Syst Appl 2024;248:123439. [CrossRef]

[34] Suresh K, Velmurugan KJ, Vidhya R, Kavitha V. Deep Anomaly Detection: A Linear One-Class SVM Approach for High-Dimensional and Large-Scale Data. Appl Soft Comput 2024;112369. [CrossRef]

[35] Mohammed B, Gbashi EK. Intrusion detection system for NSL-KDD dataset based on deep learning and recursive feature elimination. Eng Technol J 2021;39:1069–1079. [CrossRef]

[36] Kavitha S, Uma Maheswari N, Venkatesh R. Network anomaly detection for NSL-KDD dataset using deep learning. Inf Technol Ind 2021;9:821–827. [CrossRef]

[37] Wang D, Nie M, Chen D. BAE: Anomaly detection algorithm based on clustering and autoencoder. Mathematics 2023;11:3398. [CrossRef]

[38] Tian Q, Han D, Li KC, Liu X, Duan L, Castiglione AJAI. An intrusion detection approach based on improved deep belief network. Appl Intell 2020;50:3162–3178. [CrossRef]