MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3367-3375

Improved Hybrid Multi-Model Technique to identify Anomalies in Traffic Signals

Keerthi K[1], Dr. Anitha Karthi[2]
[1] Research scholar, Department of Computer Science Engineering, Bharath Institute of Higher Education and Research, Chennai India
[2] Professor, Department of Computer Science Engineering, Bharath Institute of Higher Education and Research, Chennai India
[1] keerthi2026kuppuswamy@gmail.com; [2] anithakarthicse@bharathuniv.ac.in
Corresponding Author: Keerthi K; keerthi2026kuppuswamy@gmail.com

Abstract: Anomaly detection in traffic signals is a critical component for ensuring the efficient operation of transportation systems and improving road safety. Traditional traffic signal monitoring methods rely heavily on predefined thresholds and rule-based systems, which often fail to adapt to dynamic traffic conditions or detect novel and complex anomalies. This paper explores an advanced deep learning approach for anomaly detection in traffic signals, leveraging Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based models to detect irregular patterns in traffic signal data. We propose a hybrid model that integrates both temporal and spatial features from traffic signal data, including signal status, traffic flow, and sensor inputs, to identify anomalies such as malfunctioning signals, traffic congestion, or communication failures. The model employs a multi-modal architecture that processes both time-series data (e.g., traffic volume) and spatial data (e.g., signal configurations) for more accurate anomaly detection. We also introduce a novel unsupervised anomaly detection technique based on autoencoders and attention mechanisms, enabling the model to learn from unlabelled traffic data and detect previously unseen irregularities. Extensive experiments on real-world traffic datasets demonstrate the superior performance of the proposed approach, achieving high detection accuracy with low false positives, thereby providing a robust solution for real-time traffic signal monitoring and management. This method offers significant improvements over traditional approaches by adapting to evolving traffic patterns and enhancing the resilience of smart transportation systems.

Keywords: Anomaly detection, traffic signals, deep learning, CNN, RNN, Transformer, autoencoder, attention mechanisms, unsupervised learning, smart transportation systems.

I. INTRODUCTION

Using machine learning and deep learning to analyze computer vision has become very popular in visual data computing as the need for public monitoring systems grows. The visual data offers a broader spectrum of information in comparison to other data sources such as GPS, mobile location, and radar signals. The plotted intelligence has more important features for detecting and predicting traffic congestion, accidents, and other irregularities than the statistical data that is accumulated from road traffic. Distinctive feature patterns in video sequences, differing from the normative, may be classified as anomalies. Anomaly detection is a branch of video surveillance that uses the space-time properties of similar visual objects to look for things that aren't right in video scenes [1]. Anomaly-based detection algorithms often necessitate the acquisition of normal behaviour during training to assess deviations during testing. Vehicles on walkways, crossing pedestrians when the green light is on, signal-skipping at a crosswalk, and U-turning at red lights are a few instances of abnormalities in traffic flow. Traditional techniques for identifying anomalous behaviours in surveillance systems depend on substantial human effort to scrutinize CCTV recordings continuously. New automated detection methods can speed up the process, minimize personnel expenses, and prevent human errors. Conditions such as lighting, weather, occlusion, day, and night [2] are important factors that can impair the effectiveness of visual anomaly identification in traffic videos.Since the 21st century, the internet has evolved rapidly, and modern civilization relies on network services with more terminal devices connected to the internet. An unprecedented volume of network data transmission and a more complicated network environment have created new network security issues. Maintaining network security and stability is getting increasingly difficult as malicious software, terminal platforms, and network protocols become more sophisticated, making network vulnerabilities simpler to exploit [3]. Traditional network security methods that use signature technology to detect assaults are increasingly useless against “zero-day attacks”. Network attacks that exploit vulnerabilities that software developers are unaware of prior to their discovery by attackers are referred to as zero-day attacks. Because these vulnerabilities are poorly protected, attackers can readily exploit them, causing system and data damage, data breaches, system failures, and financial losses. Signature-based network security technology adds attack signatures to the database after major attacks [4].In 2021, a Java-based remote code execution vulnerability in Apache Log4j2 was made public, impacting numerous systems and apps that use Apache Log4j2 globally [3]. A successful exploit can remotely execute arbitrary code on susceptible systems, resulting in data leakage, system crash, unauthorized access to sensitive data, and more. To address these difficulties, traffic anomaly detection has been extensively studied and used as a solution for protecting networks and detecting anomalies [4,5,6,7]. Statistics-based classification algorithms are first. These algorithms model behavioural indications using one or more variables [8]. Domain knowledge-based categorization algorithms are the second kind [9]. The final group is machine-learning classification algorithms [10]. Now, the primary research focus in anomaly detection technology is on traditional machine learning and deep learning-based anomaly detection [11].

Bhattacharya et al. [12] suggested a principal component analysis firefly-based XGBboost classification model that performed well on a Kaggle dataset. Ding et al. [13] suggested a combined intrusion detection technique based on K-nearest-neighbors (KNN) and generative adversarial networks for uneven data. Balyan et al. [14] used the enhanced genetic algorithm, particle swarm optimization (EGA-PSO), and improved random forest (IRF) methods to create an efficient detection model based on a hybrid network and obtain excellent accuracy on the NSL-KDD dataset. Ahmim et al. [15] demonstrated gains in terms of false positive rate and time cost by merging decision trees with several categorization techniques. Tao et al. [16] proposed FWP by merging the features of genetic and vector machine algorithms and creating a fitness function. Aljanabi et al. [17] suggested a better binary classification algorithm that was built on the TLBO teaching and learning algorithm and the algorithm of genetics. It works admirably on the KDDCUP dataset. Ioannou et al. [18] developed a detection system based on binary metalogic regression (BLR), which took local node parameters of benign and malicious behaviours as input and obtained attack detection accuracy ranging from 96 to 100%. Random forest method and Pearson correlation analysis helped Cao et al. [19] pick characteristics and eliminate repetition. Convolutional neural networks retrieve spatial data more accurately in similar models. LGMAD, a real-time anomaly detection system based on short-term memory and the Gaussian mixture model, was proposed by Ding et al. [20] and performed well on datasets they created themselves. Suda et al. [21] proposed RNN-based vehicular network anomaly detection. This approach can extract time-series data from network packets and identify ID spoofing and flooding attacks. Roy et al. [22] suggested using bidirectional long short-term memory recurrent neural networks (BLSTMRNNs) to find strange things in information collected by networks in the IoT. They investigated the performance of binary categorization of regular and attack traffic in the Internet of Things. In their unsupervised multivariate anomaly detection technique using generative adversarial networks (GANs), Li et al. [23] produced two new datasets to validate the model. A network intrusion detection technique utilizing a memory-augmented deep autoencoder (MemAE) was presented by Min et al. [24]. After training the MemAE model to rebuild the input of an aberrant sample near to a normal sample, the generalization problem is solved. Experimental results on the NSL-KDD, UNSW-NB15, and CICIDS 2017 datasets showed that the suggested technique outperforms other one-class models. Tab

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3367-3375

Transformer by Xu et al. [25] uses the self-attention mechanism and gradient lifting decision tree to extract discrimination features and perform well in NAD data sets.

Most research on network traffic anomaly detection uses open-source data sets and is done in numerous contexts and conditions. Finally, the model based on the short- and long-term memory networks effectively preserves the long-term reliance link between traffic data. An autoencoder-based model may process data and classify it. GAN-based models are better for small sample training. A transformer-based model can more thoroughly extract the context of the data and precisely capture the anomalous link between the data by utilizing the self-attention mechanism.Examines the building of a traffic anomaly detection dataset from network packets to facilitate algorithm training. The current classic dataset is either older or lacks full feature extraction, while open-source feature extraction algorithms contain mistakes and cannot extract new features. Feature extraction program was re-developed on the Linux platform, and a more comprehensive set of features was created by combining domain knowledge and various machine learning algorithms, so that abnormal traffic types can be distinguished using this set of features and then feature elimination can be performed to reduce redundancy between features. Experiments demonstrated that the complete feature-processing procedure presented in this article can lower the algorithm's time and spatial complexity while also improving its performance.

- A network model using improved RNN and self-attention was proposed second. Network traffic data are time series data, and RNN, one of the strongest benchmark models for solving sequence problems in deep learning, has some issues, such as its long-term dependence on old data during prediction.
- By including short- and long-term memory networks with gated recurrent neural networks, we may address the issues of gradient disappearance and explosion in RNN.
- An N-layer enhanced RNN model learns temporal information in network traffic data, but the sequence's relevance varies.
- The self-attention system has sequence components.

The suggested network model outperformed the benchmark model on the dataset to prove its efficacy.

- Creating a feature-based engineering dataset
- Detecting anomaly network traffic using self-attention networks
- Summarizing prior material and prospects.

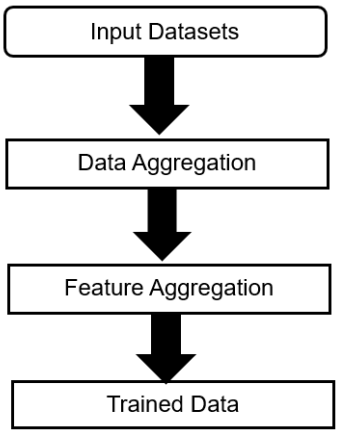


Figure 1. Data Processing

II. FEATURE EXTRACTION PROCESS

A. Dataset Creation based on Feature Engineering:

To teach and assess detection of intrusions, the US department's Protection Advanced Planning Agency replicated the Air Force's network in 1998. This dataset became the KDDCUP99 dataset [26] after preprocessing, providing a performance benchmark for future network traffic anomaly detection algorithms. The KDDCUP99 dataset is still rather rough, nevertheless, with a significant variance in the proportion of normal and abnormal data as well as a lot of redundant and duplicate data. To address these concerns, Tavallaee et al. upgraded KDDCUP99 to create NSL-KDD [27]. After entering the twenty-first century, the popularity of network technology has expanded dramatically, and the rapid expansion of infrastructure and software facilities has resulted in major developments in the network environment. Due to the age of datasets like NSL-KDD, network traffic distribution differs from previous years. As assaults evolve, hardware technology makes it easier to capture data and information. Datasets with deeper traffic characteristics and more acceptable data distribution are continually emerging, such as CICDS2017 [28]. DDoS 2019 [29] and distributed denial-of-service datasets are also available. Different fields will generate their own traffic anomaly detection datasets based on the business they manage, but these results will be confidential.To teach and assess detection of intrusions, the US department's Protection Advanced Planning Agency replicated the Air Force's network in 1998. This dataset became the KDDCUP99 dataset [26] after preprocessing, providing a performance benchmark for future network traffic anomaly detection algorithms. The KDDCUP99 dataset is still rather rough, nevertheless, with a significant variance in the proportion of normal and abnormal data as well as a lot of redundant and duplicate data. To address these concerns, Tavallaee et al. upgraded KDDCUP99 to create NSL-KDD [27]. After entering the twenty-first century, the popularity of network technology has expanded dramatically, and the rapid expansion of infrastructure and software facilities has resulted in major developments in the network environment. Due to the age of datasets like NSL-KDD, network traffic distribution differs from previous years. As assaults evolve, hardware technology makes it easier to capture data and information. Datasets with deeper traffic characteristics and more acceptable data distribution are continually emerging, such as CICDS2017 [28]. DDoS 2019 [29] and distributed denial-of-service datasets are also available. Different fields will generate their own traffic anomaly detection datasets based on the business they manage, but these results will be confidential.Network attack detection solutions are tested using UNSW-NB15 [30]. UNSW's School of Computer Science provides it. This dataset comprises a lot of simulated network traffic data, including DoS, Trojan, and malware attacks. The dataset contains 49 features, the majority of which are technical indicators of network traffic protocols, such as packet size and time interval. This dataset is popular in network security research and used to test methods. These properties don't adequately characterize network traffic. DoS attacks are identified by the frequency of sending data packets quickly, SYN flooding attacks by the number of SYN in the TCP header, and XSS injection by the amount of data difference between packets in a session. The UNSW-NB15 dataset lacks several crucial properties, limiting algorithm detection and the computation of various statistical indicators to the UNSW-NB15 feature set, rebuild it, and create a dataset to increase the algorithm's detection performance.

B. PERFORMING FEATURE EXTRACTION:

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3367-3375

Extracting relevant features from raw data for analysis and processing is feature extraction. Machine learning algorithms utilize feature extraction to select meaningful information from raw data to help them find patterns. To thoroughly and accurately define network traffic aspects, this article designs five kinds of features based on the retrieved binary session data as represented in table 1.

Table 1. Feature Types

Feature Category	Quantity
Identity features of conversation	5
Transmission features of the session	38
Frequency features of session	19
The repetitive features of session	19
TCP session features	27

Algorithm 1: Feature set correlation

Input: Weighted Feature Set, Feature Correlation Set, Weighted Set.
Output: Select feature set correlation value 'FS'.
Initialize the Weighted Set assigned to the feature set, F_g, Feature 'F' along with the correlation factor of minimum and maximum value.
Calculate the Feature Set (FS) as it evaluates the processing of weighted feature set.

```
Check all the Feature Set Correlation value
{
        Select if feature correlation value is maximum.
        Else
        Deselect if feature correlation value is less
}
```

Based on the algorithm 1, Feature Set 'FS' is evaluated based on the process of Weighted Feature Set 'Wf(F)'. To process 'FS', consider the maximum feature correlation as zero otherwise make the feature value gets avoided since it as less weight. The dataset contains features with weight is considered and it is analysed to detect the detection process. Hence, the features are loaded and processed by analysing 'FS'.

Algorithm 2: Feature set Normalization

Input: Weighted Feature Set
Output: Feature Normalization 'FN'
Initialize the feature set value 'FS' form feature set correlation algorithm
To calculate the feature normalization based on features.
Check whether features values are same for both features,
Then alter the feature values.
Do
{
        Identify the mean values for the features to balance the feature values.
        Calculate the standard deviation among the relevant feature set.
}
Based on the algorithm 2, output of the algorithm 1 'FS' is considered as the input to analyse and validate the features to obtain the Feature Normalization 'FN'. By considering the median and standard deviation, Feature Set 'FS' is being analysed. $m_g()$ is used to find the mean values of the features so that normalization may be accomplished by balancing the feature values, and std is used to calculate the standard deviation of the relevant feature set.

Algorithm 3: Feature Batch generation

Input: Feature Normalization; Feature Set
Output: Generating Feature Batch
Initialize Feature Set Correlation 'FS' from Algorithm 1
        Feature Set Normalization 'FN' from Algorithm 2.
To generate feature batch generation as follows,
Batch Feature Generation → Identify the information by batch generators.
        Data Collect → transaction information and down sampling (Data Source)
Get
{
    Batch Size 'λ' based on feature set
    Batch Vectors 'γ' based on maximum normalized features.
}
Based on the Features and Correlation,
{
        Formulate Feature Normalization 'FN' based on Batch Size.
        Formulate feature vector 'FV' limit as maximum Feature Normalization 'FN' maximum with the feature incremented with 1 along with median estimation of Feature Set 'FS" maximum for the feature.
}
With the input of Feature Normalization 'FN', Feature Batch 'FB' is generated as based on the feature 'F' and Correlation limit. Here, λ → Batch size based on the normalized feature set & γ → Handle batch vectors of maximum-normalized features in algorithm 3.
Batch generators allow batch feature generation, which can be used to extract data features. They allow data collection using transaction information and down sampling, depending on the source. Batch Feature Perspectives can materialize new feature sets automatically.

Algorithm 4: Enhanced Logistics (EL_{reg}) Model Analysis.

Input: Feature Batch Generation (FB); Feature Normalization (FN); Feature Set (FS); Batch Size 'λ';
Output: Logistics 'L_{reg}' & Enhanced Logistics 'EL_{reg}'
Initialize Feature Batch Generation (FB); Feature Normalization (FN); Feature Set (FS); Batch Size 'λ'.
To calculate the logistics regression 'L_{reg}' as it utilizes the single dependent variable by considering two values.
Maximum 'Single independent variable 'S'
Correlation variable 'F' and increment the feature 'F+1'
ADD
Perform logarithmic on Maximum value of the Feature Batch Generation (FB) along with Batch Size 'λ'.
Analyse 'EL_{reg}' as it takes individual variables as it takes both batch and multi-level independent feature set.
        Summation with the limit from Feature 'F' from 1 to m
        {

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3367-3375

Min (Feature Batch Generation 'FB' on incremented 'F+1')
}
Summation with the limit from Feature 'F' from 1 to m
{
Based on the maximum correlation value,
{
Add incremented Feature 'F' with Diff (Messgae 'M', Feature 'F')
Divide
Logistics 'L_{reg}' (Feature)
}
Maximum Batch Size 'λ_{max}' on Logistics 'L_{reg}' applied on above formulation

In Algorithm 4, Enhanced Logistics method figures out how likely something is to happen by assuming that the log odds of it are linear combinations of one or more independent factors. Logistic regression estimates variables in a logistic model in a regression investigation. In logistic regression, a single dependent variable with two possible values (zero and one) is used, and the explanatory variables can be discrete binary or continuous. ELR analyzes attributes and relates them to the batch feature set and generates multi-level independent feature set.

Algorithm 5: Feature Relations 'F_{rel}' based on Mean 'M' and message 'm;

Input: Feature 'F'; Enhanced Logistics 'EL_{reg}';
Output: Feature Relations 'F_{rel}'; Unified Feature Relations 'UF_{rel}'
Initialize Feature 'F' & Enhanced Logistics 'EL_{reg}'
To formulate Feature Relations 'F_{rel}' for Mean 'M'
Summation with the limit from Feature 'F' from 1 to m
{
Minimum Diff for (Feature 'F' & Increment Feature 'F+1')
Standard Deviation (Feature (EL_{reg})
}
Formulate UFrel[m] (Unified Feature Relations)
{
Summation with the limit from Feature 'F' from 1 to R
F_{rel} (Feature 'F' with the incremented Feature 'F) - Minimum Diff for (Feature 'F' & Increment Feature 'F+1')
Divide
Summation with the limit from Feature 'F' from 1 to m
($\gamma \rightarrow$ Batch Size) Multiply Size (Feature Batch (FB))
}

From the Algorithm 5, the correlation-based feature processing method's filter-like nature allows it to be utilized with any detection model. It examines feature subsets using only data's inherent features to detect fraud. Popular for quantifying the connection between two feature set traits is the feature relation coefficient. Linearly dependent qualities have a relationship coefficient of 1. No relationship between features is indicated by a relational coefficient of 0. Feature relations are created.

Algorithm 6: Model Prediction

Input: Feature Relations 'F_{rel}'; Unified Feature Relations 'UF_{rel}'
Output: Model Prediction '$PSet$'
nitialize Feature Relations 'F_{rel}'; Unified Feature Relations 'UF_{rel}'; Feature 'F'
Summation with the limit from Feature 'F' from 1 to M
{
Feature Difference d_f (UF_{rel})
Maximum Value (F_{rel} (Feature 'F' & Increment Feature 'F+1'))
}
Add
Summation with the limit from Feature 'F' from 1 to M
{
Batch Size 'γ' (UF_{rel}(Feature 'F')) + Summation with the limit from Feature 'F' from 1 to N (F_{rel} (Increment Feature 'F+1')
Divide
S_z(Unified Feature Relations 'UF_{rel}')
}
Model Prediction '$PSet$' on Unified Feature Relations 'UF_{rel}' with 'M'

In the algorithm 6, which is used to perform prediction process based on the proposed model as it takes the input of Feature Relations 'F_{rel}' and Unified Feature Relations 'UF_{rel}'. As the features are limited from 1 to M, Feature Difference d_f (UF_{rel}) is determined and add those values with batch size based on features assigned with the unified feature relation.

III. MODRES_UNet Model

The proposed $IMPRS$_UNet model architecture integrates residual and wide blocks (Fig. 2). This network architecture employs local and global feature extraction. The use of a 256 × 256 image resolution for output image generation demonstrates superior performance analysis. The residual network integrates the $IMPRS$_UNet backbone with additional residual blocks within a deep framework to address gradient mitigation. In this $IMPRS$_UNet design, the convolutional neural network is responsible for image segmentation into three parts: encoder, decoder, and contraction and expansion. Convolutional 2D, Max pooling 2D, Wide Context, Transpose 2D, Residual Block, and Concatenate metrics are used to widen blocks in the suggested architecture.

The current U-Net architecture executes the following convolutional operations: Max pooling, activation, and layers of sampling.

The sample layer is categorized into contraction, bottleneck, and expansion. Blocks of contraction are used as input in the architecture, which also has a 3x3 convolutional layer and 2x2 max pooling. In the pooling layer, feature mapping is doubled. Subsequently, 3x3 convolutional layers and a 2x2 up-convolution layer are implemented at the layer that is causing a slowdown. Expanding blocks utilize 3x3 convolutional layers and 2x2 upsampling layers as input, incorporating feature channels. Concatenation and feature mapping result in a constricted path. The 1x1 convolutional layer generates feature mappings based on segments. U-Net analyzes images through a loss function to identify cellular structures based on segmentation.

i. Max Pooling layer:

Maximum pooling has been integrated with average pooling, resulting in a new layer that facilitates down feature extraction mapping. This demonstrates the feature patches derived from the feature map method. Max pooling computes the average presence derived from feature

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3367-3375

extraction and activates relevant feature information. Neuronal network-based convolutional layer, filtered learning on input images, extracts mapping information. The layer exhibits enhanced performance in low-level feature extraction. Features are represented in the convolutional layer.

ii. Dropout Layer:

Overfitting in convolutional neural networks is prevented by this technique. If the hidden layer's neurons are set to 0 during training phase updates, dropout will concentrate on settings. Below, we discuss the Residual U-Net numerically and its sigmoid-shaped activation function.

$$IMPRS\_UNet\ (I) = \begin{cases} 0 \ if\ I \leq 0 \\ I\ otherwise \end{cases}$$

$$Sh\ (M) = \frac{1}{1 + Exp\ (-I)}$$

Algorithm 7: Modified Max pooling in Convolutional Neural Networks

Input: Images

Output: Calculate the Max Pooling

1. Initially the max pooling operation is applied to output mapping feature using convolutional operations detector line.
2. For the given stride, max operation is calculated by specifying

Max (3.0, 3.0)

3. Max pooling operation is added with MaxPooling2D

Sequential pattern

Relu activation function are summarized.

5. Vertical line detector is defined
6. Store the weight in the max pooling model
7. Apply the data filter

Calculate the max / largest value of mapping feature for each patch.

Calculate the Max pooling.

In the modified max pooling algorithm of the CNN, images are utilized as input. Max pooling operations are initialized and applied to construct the output mapping utilizing convolutional operation detector line. These operations are based on the pictures for which they are applied. To determine max operations, it is necessary to specify the decimal values for the picture stride that is been provided. Two-dimensional convolution is then used to add the max operations to the total. It defines the sequential pattern and the activation function 'Relu' is used. After 2D, define a vertical line detector and store the model value in the model. Following that, the data filter is used to determine the highest possible value of feature mapping for each individual patch separately. The maximum pooling is computed using the procedure described above.

Algorithm 8: Modified Dropout Layer in Convolutional Neural Networks

Input: Image Datasets

Output: Perform Dropout model

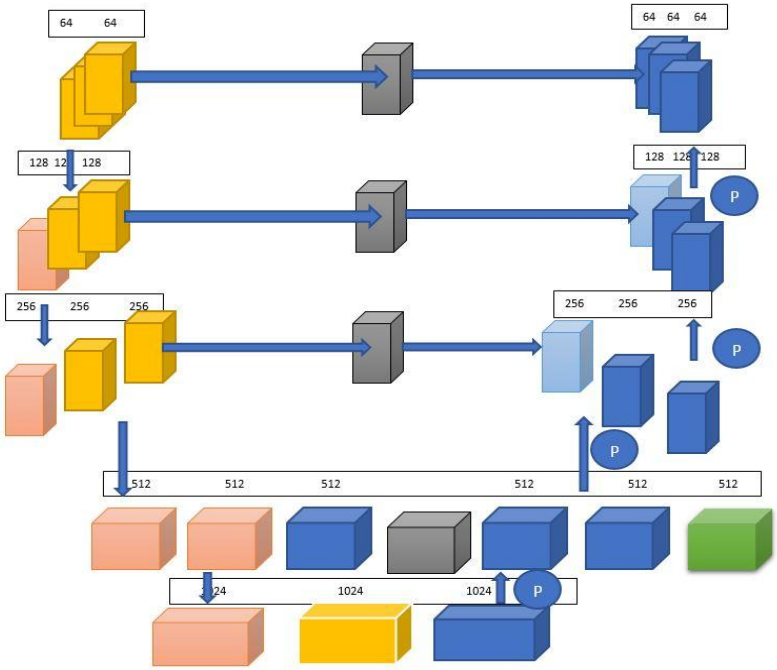
1. Use keras to construct the neural network based on Dense and Flatten layer with Activation function.
2. Import & load the data and it is spitted into training and testing sets.
3. Show the training data using cmap with the binary conversion.
4. Perform Training and Testing by normalizing and categorical

Sequential Layer

Flatten / Dense Layer

Activation model

5. The dropout model is performed



→ Modified A_t

Figure 2. Architecture of IMPRS_Unet Model

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3367-3375

The algorithm2 is a modified dropout layer-based CNN that accepts picture datasets as inputs. Keras employs a dense and flatten layer with an activation function to build a neural network. Import and load the data when the neural network has been constructed, and then divide the data into two groups according to the training and testing procedures. The data is then displayed as a binary conversion using cmap. Prior to the process of training and testing based on normalization and categorization, in addition to dense and flatten layers with activation function. At long last, the dropout model is developed and put into application.

iii. A_t Activation Model

The brain tumour variation in the image dimensionality is taken into account for a variety of complex structures, and there are small-scale tumours. If you want to get useful information, the network will have more features with down sampling. High-level mapping is used to obtain convolutional and data transformation through non-linearities. This mapping is done with spatial information and location. Additionally, the attention model contributes to the improvement of the data segmentation performance accuracy for the brain tumour that is 1.5 to 2 centimetres in size. In this approach, attention gates and residual models are combined to form the Res U-Net model. The attention gate is a tool that assists in the utilization of spatial and location information that is based on low-scale mapping features through the process of up sampling. This is done in order to handle the issue of changing dimensions and complicated structures. The suggested Res U-Net architecture, the attention model, examines the small scale tumour to execute the picture segmentation task well, as shown in Figure 2 & 3, when combined with the (p) Modified At Model, as shown in Figure 24. Right here Features mapping with a 'L' layer and a gi signal gate are referred to as XL. After this, the attention coefficient is established within the bounds of 0 to 1, and the results are obtained through element-wise addition and multiplication. After this, the multi-dimensional attention coefficient for 'i' can be determined by utilizing the sigmoid function and relu.

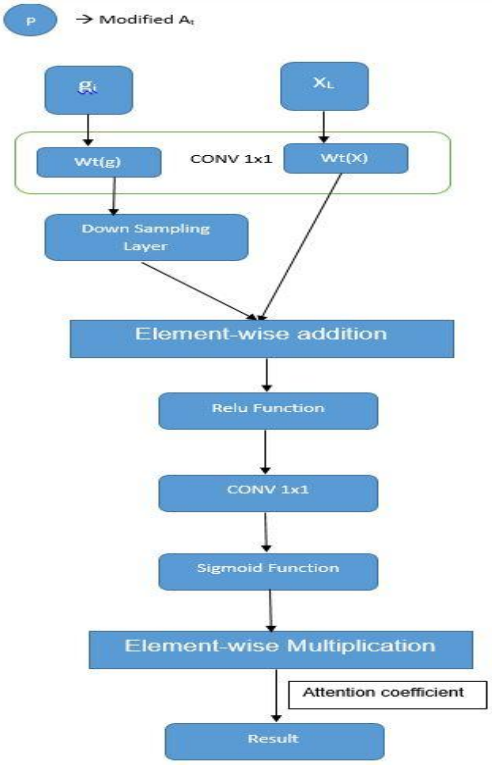


Figure 3. Modified A_t Model

Dense information feature extraction employs down sampling, which can restore spatial and location-specific information through up sampling. A representation of the updated residual block may be found in Figures 2 and 3.

iv. Wide Context Information

In a broad context block, two parallel connections are used as input for two convolutional layers. Both 1 x N and N x 1 filters are recommended for usage in the first connection. Subsequently, the connection employs a convolutional model with a 1 x N filter and a N x 1 filter. In this process, the information is extracted to establish data similarity by analyzing the broad context. It is subsequently classified into various convolutional subclass networks for brain tissue, as illustrated in Figure 3.

IV. PERFORMANCE ANALYSIS

i. FEATURE EXTRACTION:

The original data for feature extraction in this article comes from the UNSW-NB15 dataset's pcap packets. After processing with the feature-extraction program in this article, a dataset of 2,445,079 records is obtained. The experimental data results.

First, classic machine learning models are trained on the dataset as a benchmark performance. The dataset is divided into a training set, validation set, and test set in an 8:1:1 ratio. Then, linear classifiers, decision trees, random forests, LightGBM, XGBoost, and multilayer perceptron are trained and tested, and the results are shown in the Table 2 and 3.

Table 2. Machine learning algorithm training results.

Existing Algorithms		Linear Classification	Decision Tree	Random Forest	Light GBM	XG Boost	Multi-Layer Perceptron
Training Set	Accuracy	0.77	0.99	0.99	0.86	0.88	0.85
	Precision	0.32	0.96	0.98	0.56	0.51	0.46
	Recall	0.44	0.99	0.98	0.73	0.85	0.74
	F1-Score	0.36	0.98	0.98	0.63	0.64	0.57

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3367-3375

Table 3. Machine learning algorithm testing results.

Existing Algorithms		Linear Classification	Decision Tree	Random Forest	Light GBM	XG Boost	Multi-Layer Perceptron
Testing Set	Accuracy	0.77	0.83	0.87	0.86	0.87	0.85
	Precision	0.31	0.60	0.55	0.55	0.49	0.45
	Recall	0.44	0.60	0.78	0.72	0.84	0.74
	F1-Score	0.35	0.60	0.65	0.62	0.62	0.56

When the number of features in the subset was 19, the algorithm achieved the best F1-score of 0.6272. Therefore, 19 features were selected as the final optimal subset from the original 121 features.The dataset was then split into training, validation, and testing sets with a 8:1:1 ratio.The experimental results are shown in the Table 6. To evaluate the impact of the feature subset on algorithm performance, the model was retrained using only the 19 selected features, and the testing results were compared to those using all features as shown in Table 4. It can be observed that the random forest algorithm still achieved the best F1-score of 0.6786, which is 2.1% higher than before feature selection. This indicates that feature selection effectively improves the overall performance of the algorithm.

Table 4. Machine learning algorithm training and test results before and after feature selection.

Algorithms	Linear Classification	Decision Tree	Random Forest	Light GBM	XG Boost	Multi-Layer Perceptron
Before Feature Extraction	0.36	0.60	0.65	0.62	0.62	0.56
After Feature Extraction	0.40	0.60	0.67	0.63	0.63	0.51

ii. PROPOSED NEURAL MODEL:

In the performance analysis, effectiveness can be determined based on the number of images deliberated. The TensorFlow deployed with specific modules as learning rate scheduler and checkpoint. Kaggle dataset contains specific attributes such as patient, RNSseq cluster, laterality, tumor location, and ethnicity. Then those datasets contain images and perform sorting the data with checkpoint and lead to final datasets. The final dataset contains patient id, image path and mask path as mentioned in Table 5. To determine the mask count plot by comparing the count value and mask.

Table. 5 Mask Count Part

S.No.	Mask	Dtype
1	0	2556
2	1	1373

The brain tumor images are varied based on the pixel colour to identify the position of the tumor on mask value. Then a new dataset is created based on the variables such as image_path, mask_path and mask. Based on the model selection, trained data are tested and separated based on brain_df_train, test_size = 0.15. Then those images are validated.

The images are trained based on the classifier model concerning Residual Network ResNet50 is being used. The input layer performs image classification on Conv2D, Input, ZeroPadding2D, BatchNormalization, Activation, MaxPooling2D, batch normalization images. The classifier model obtained a data loss of 0.2353 and a data accuracy of 94.745% based on the image classification model to detect whether the tumor exists or not. The accuracy_score, confusion_matrix, and classification_report metrics create the confusion matrix based on original and prediction images as represented in Table. 6 and 7.

Table. 6. Attribute metrics based on mask value

Mask_Value	Precision	Recall	F1-Score	Support
0	0.93	0.99	0.96	382
1	0.98	0.87	0.92	208

Table. 7. Attribute metrics based on mask value (Accuracy; Macro Avg; Weighted Avg)

Mask_Value	Precision	Recall	F1-Score	Support
Accuracy	-	-	0.95	590
Macro Avg	0.96	0.93	0.94	590
Weighted Avg	0.95	0.95	0.95	590

In Figure. 4, the loss is calculated based on the classification model by varying the epochs from 0 to 30. The loss gets saturated as the epochs increase in the value for the proposed Hybrid Multi-Model Technique.

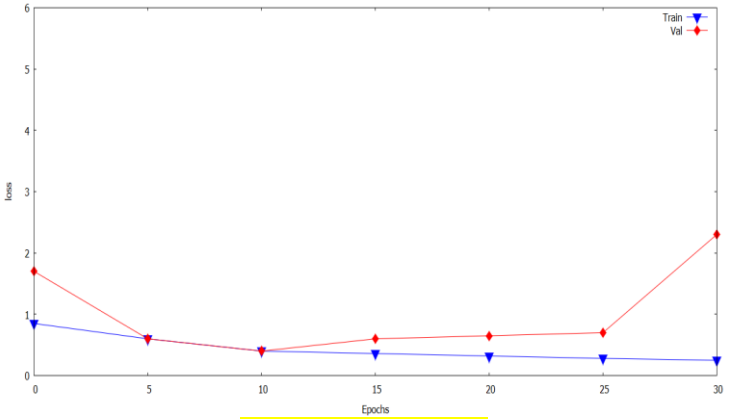


Figure 4. Epochs Vs Loss

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3367-3375

The Accuracy is calculated and increases based on the increase in the epochs value, which varies from 0 to 30. It is represented in Fig, 5.

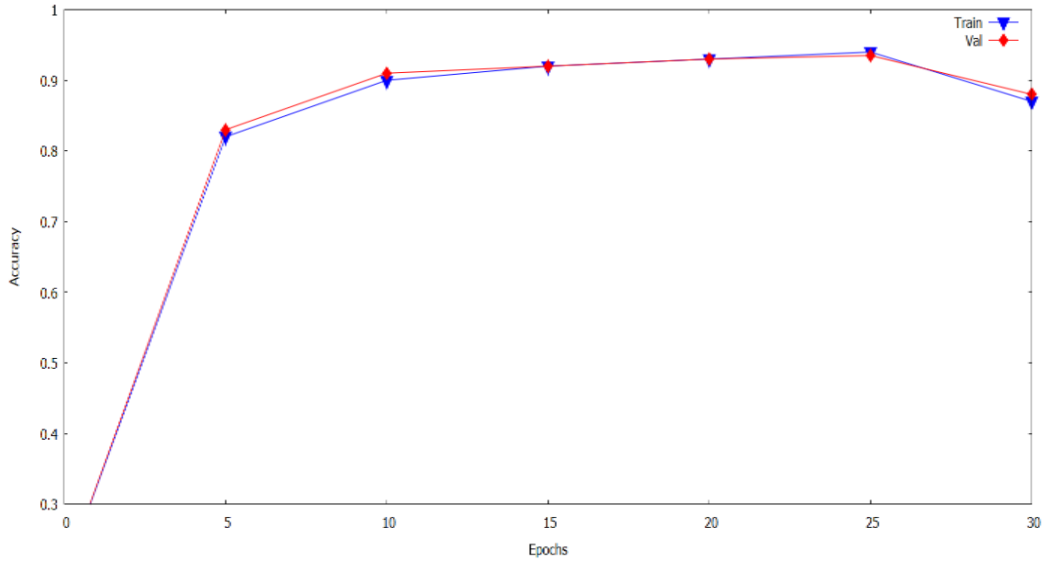


Figure 5. Epochs Vs Accuracy

V. CONCLUSION

investigates a sophisticated deep learning method for the detection of anomalies in traffic signals. This method makes use of Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based models to identify unusual patterns in traffic signal data. Our hybrid model uses temporal and geographical variables from traffic signal data, including signal status, traffic flow, and sensor inputs, to identify abnormalities such signal malfunctions, traffic congestion, and communication failures. A multi-modal architecture is utilized by the model, which processes both time-series data (for example, traffic volume) and geographical data (for example, signal configurations) to achieve a higher level of accuracy in the detection of anomalies. Furthermore, we present an innovative unsupervised anomaly detection technique that is founded on autoencoders and attention processes. This technique enables the model to acquire knowledge from unlabeled traffic data and identify deviations that were not previously observed. Based on the existing algorithms, such as linear classifiers, decision trees, random forests, LightGBM, XGBoost, and multilayer perceptron, which are trained and evaluated, extensive experiments have been conducted on real-world traffic datasets. These studies have proved the effectiveness of these methods. Feature selection appears to be an excellent method for improving the overall performance of the algorithms. Also, attribute measures based on mask value are compared to checking the accuracy, recall, F1 Score, and support. These include Accuracy, Macro Avg, and Weighted Avg. Finally, the proposed Hybrid Multi-Model Technique is evaluated using loss and accuracy values depending on the number of epochs.

REFERENCES:

1. Aboaoja, F.A.; Zainal, A.; Ghaleb, F.A.; Al-rimy, B.A.S.; Eisa, T.A.E.; Elnour, A.A.H. Malware Detection Issues, Challenges, and Future Directions: A Survey. *Appl. Sci.* 2022, *12*, 8482. [Google Scholar] [CrossRef]
2. Liaropoulos, A. Cyber-Security: A Human-Centric Approach. In *European Conference on Cyber Warfare and Security*; Academic Conferences International Limited: Oxford, UK, 2015; p. 189. [Google Scholar]
3. Juvonen, A.; Costin, A.; Turtiainen, H.; Hamalainen, T. On Apache Log4j2 Exploitation in Aeronautical, Maritime, and Aerospace Communication. *IEEE Access* 2022, *10*, 86542–86557. [Google Scholar] [CrossRef]
4. Ferrag, M.A.; Shu, L.; Friha, O.; Yang, X. Cyber Security Intrusion Detection for Agriculture 4.0: Machine Learning-Based Solutions, Datasets, and Future Directions. *IEEE Caa J. Autom. Sin.* 2022, *9*, 407–436. [Google Scholar] [CrossRef]
5. Hussain, F.; Abbas, S.G.; Shah, G.A.; Pires, I.M.; Fayyaz, U.U.; Shahzad, F.; Garcia, N.M.; Zdravevski, E. A Framework for Malicious Traffic Detection in IoT Healthcare Environment. *Sensors* 2021, *21*, 3025. [Google Scholar] [CrossRef] [PubMed]
6. Shi, L.Y.; Wu, L.F.; Guan, Z.T. Three-layer hybrid intrusion detection model for smart home malicious attacks. *Comput. Electr. Eng.* 2021, *96*, 107536. [Google Scholar] [CrossRef]
7. Yang, L.; Moubayed, A.; Shami, A. MTH-IDS: A Multitiered Hybrid Intrusion Detection System for Internet of Vehicles. *IEEE Int. Things J.* 2021, *9*, 616–632. [Google Scholar] [CrossRef]
8. Ye, N.; Emran, S.M.; Chen, Q.; Vilbert, S. Multivariate statistical analysis of audit trails for host-based intrusion detection. *IEEE Trans. Comput.* 2002, *51*, 810–820. [Google Scholar] [CrossRef]
9. Viinikka, J.; Debar, H.; Mé, L.; Lehikoinen, A.; Tarvainen, M. Processing intrusion detection alert aggregates with time series modeling. *Inf. Fusion* 2009, *10*, 312–324. [Google Scholar] [CrossRef]
10. Buczak, A.L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Commun. Surv. Tutor.* 2016, *18*, 1153–1176. [Google Scholar] [CrossRef]
11. Halbouni, A.; Gunawan, T.S.; Habaebi, M.H.; Halbouni, M.; Kartiwi, M.; Ahmad, R. Machine Learning and Deep Learning Approaches for CyberSecurity: A Review. *IEEE Access* 2022, *10*, 19572–19585. [Google Scholar] [CrossRef]
12. Bhattacharya, S.; Maddikunta, P.K.R.; Kaluri, R.; Singh, S.; Gadekallu, T.R.; Alazab, M.; Tariq, U. A Novel PCA-Firefly Based XGBoost Classification Model for Intrusion Detection in Networks Using GPU. *Electronics* 2020, *9*, 219. [Google Scholar] [CrossRef]
13. Ding, H.; Chen, L.; Dong, L.; Fu, Z.; Cui, X. Imbalanced data classification: A KNN and generative adversarial networks-based hybrid approach for intrusion detection. *Future Gener. Comput.-Syst.-Int. J. Escience* 2022, *131*, 240–254. [Google Scholar] [CrossRef]
14. Balyan, A.K.; Ahuja, S.; Lilhore, U.K.; Sharma, S.K.; Manoharan, P.; Algarni, A.D.; Elmannai, H.; Raahemifar, K. A Hybrid Intrusion Detection Model Using EGA-PSO and Improved Random Forest Method. *Sensors* 2022, *22*, 5986. [Google Scholar] [CrossRef] [PubMed]
15. Ahmim, A.; Maglaras, L.; Ferrag, M.A.; Derdour, M.; Janicke, H. A Novel Hierarchical Intrusion Detection System based on Decision Tree and Rules-based Models. In Proceedings of the 15th Annual International Conference on Distributed Computing in Sensor Systems (DCOSS), Santorini, Greece, 29–31 May 2019; pp. 228–233. [Google Scholar]
16. Tao, P.Y.; Sun, Z.; Sun, Z.X. An Improved Intrusion Detection Algorithm Based on GA and SVM. *IEEE Access* 2018, *6*, 13624–13631. [Google Scholar] [CrossRef]

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 1s, 2026, Pages: 3367-3375

17. Aljanabi, M.; Ismail, M. Improved Intrusion Detection Algorithm based on TLBO and GA Algorithms. *Int. Arab. J. Inf. Technol.* 2021, *18*, 170–179. [Google Scholar]
18. Ioannou, C.; Vassiliou, V.; Association for Computing Machinery. An Intrusion Detection System for Constrained WSN and IoT Nodes Based on Binary Logistic Regression. In Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), Alicante, Spain, 28 October–2 November 2018; pp. 259–263. [Google Scholar]
19. Cao, B.; Li, C.; Song, Y.; Qin, Y.; Chen, C. Network Intrusion Detection Model Based on CNN and GRU. *Appl. Sci.* 2022, *12*, 4184. [Google Scholar] [CrossRef]
20. Ding, N.; Ma, H.; Gao, H.; Ma, Y.; Tan, G. Real-time anomaly detection based on long short-Term memory and Gaussian Mixture Model. *Comput. Electr. Eng.* 2019, *79*, 106458. [Google Scholar] [CrossRef]
21. Suda, H.; Natsui, M.; Hanyu, T. Systematic Intrusion Detection Technique for an In-Vehicle Network Based on Time-Series Feature Extraction. In Proceedings of the 48th IEEE International Symposium on Multiple-Valued Logic (ISMVL), Linz, Austria, 16–18 May 2018; pp. 56–61. [Google Scholar]
22. Roy, B.; Cheung, H. A Deep Learning Approach for Intrusion Detection in Internet of Things using Bi-Directional Long Short-Term Memory Recurrent Neural Network. In Proceedings of the 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, Australia, 21–23 November 2018; pp. 57–62. [Google Scholar]
23. Li, D.; Chen, D.; Jin, B.; Shi, L.; Goh, J.; Ng, S.K. MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In Proceedings of the 28th International Conference on Artificial Neural Networks (ICANN), Munich, Germany, 17–19 September 2019; pp. 703–716. [Google Scholar]
24. Min, B.; Yoo, J.; Kim, S.; Shin, D.; Shin, D. Network Anomaly Detection Using Memory-Augmented Deep Autoencoder. *IEEE Access* 2021, *9*, 104695–104706. [Google Scholar] [CrossRef]
25. Xu, X.; Zheng, X. Hybrid Model for Network Anomaly Detection with Gradient Boosting Decision Trees and Tabtransformer. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Virtual, 6–12 June 2021; pp. 8538–8542. [Google Scholar]
26. Bay, S.D.; Kibler, D.; Pazzani, M.J.; Smyth, P. The UCI KDD archive of large data sets for data mining research and experimentation. *ACM SIGKDD Explor. Newletter* 2000, *2*, 81–85. [Google Scholar] [CrossRef]
27. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, Canada, 8–10 July 2009; pp. 1–6. [Google Scholar]
28. Beigi, E.B.; Jazi, H.H.; Stakhanova, N.; Ghorbani, A.A. Towards effective feature selection in machine learning-based botnet detection approaches. In Proceedings of the 2014 IEEE Conference on Communications and Network Security, Xi'an, China, 30 May–3 June 2014; pp. 247–255. [Google Scholar]
29. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. In Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019; pp. 1–8. [Google Scholar]
30. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6. [Google Scholar]