International Journal of Inventive Engineering and Sciences (IJIES)
ISSN: 2319-9598 (Online), Volume-13 Issue-7, July 2026

ViTfuse_GNN: Vision Transformer-based fusion with Graphical Neural Network for detection of Intrusions in Multimedia Healthcare

Yashaswini K, Sharath M N

Abstract: ***The exponential growth of network traffic and user data has hindered the efficacy and responsiveness of network intrusion detection systems. Intrusion systems are essential in e-healthcare since they ensure the security, confidentiality, and accuracy of patients' medical information. Diagnosis and treatment mistakes may result from any alteration to the patient's real data. Traditional intrusion detection methods often fail to fully exploit the heterogeneous nature of multimedia healthcare data, which may include medical images, patient records, and network traffic metadata. Hence, this work proposes ViTfuse_GNN, a novel hybrid framework that integrates Vision Transformer (ViT)-based multimodal feature fusion with Graph Neural Networks (GNN) for robust intrusion detection. The proposed model first employs a Vision Transformer to fuse high-level semantic features from medical images, videos, and textual metadata. Experimental evaluation on benchmark multimedia healthcare intrusion datasets demonstrates that ViTfuse_GNN outperforms state-of-the-art intrusion detection models with 99% accuracy, 98% precision, 99% recall, and 97% F1-score.***

Keywords- Intrusion detection, Multimedia, Neural Network, Vision Transformer, Healthcare, VGG-19.

Nomenclature
GNN: Graph Neural Networks
DIDS: Deep Intrusion Detection System
HPS: Hyperparameter Search

I. INTRODUCTION

Intrusion detection is arduous due to the continual evolution of assaults driven by emerging technologies, frameworks, and software. In 2020, cyber-attacks surged by 17%, with 77% classified as targeted assaults, mostly aimed at personal data and passwords. Attacks on companies aimed primarily at collecting personal information [1]. These measures provide light on modern cyber-attack detection and prevention. Most hospitals use e-healthcare to quickly meet patient needs as the healthcare industry grows [2]. Hospitals must keep EHRs and Patient Records or Personal Health Records because they include vital medical information for diagnosis and treatment [3].

Manuscript received on 03 June 2026 | First Revised Manuscript received on 10 June 2026 | Second Revised Manuscript received on 07 July 2026 | Manuscript Accepted on 15 July 2026 | Manuscript published on 30 July 2026.
*Correspondence Author(s)
Yashaswini K*, Department of Computer Science and Engineering, Navkis College of Engineering, Thimanahally Hassan, Karnataka, India, Email ID: Yashaswinikphd@outlook.com
Dr. Sharath M N, Department of Artificial Intelligence and Machine Learning, Rajeev College of Engineering, Hassan, Karnataka, India. Email ID: sharathmn64@gmail.com

IoT advancements have led to a spike in smart medical devices and systems. Edge devices may retain patient data, which must be secure and accurate. Any adjustment or distortion of these facts might lead to misdiagnosis and treatment, perhaps causing death [4]. Thus, healthcare systems need a cutting-edge Hybrid Intrusion Detection System to ensure cyber-safety. IDS began as software [5]. Due to real-time needs, specialized technology was designed to monitor the network for illegal activity or policy violations. These devices' management systems report dangerous behaviors, violations, security information, and event logs [6]. Some intrusion detection systems (IDS) may prevent invasions. To protect the enterprise, the Intrusion Detection System (IDS) analyzes network traffic and system events to detect unauthorized computer access [7]. An Intrusion Detection System (IDS) collects data from a network environment, filters out unnecessary information, and determines whether the behaviour is normal or intrusive [8][9]. Researchers use data mining, soft computing, machine learning, statistics, Bayesian techniques, artificial neural networks, and evolutionary computing. Whether an activity is normal or invasive is demonstrated by network anomaly detection. Decision Tree, Support Vector Machine, and Naive Bayes are common machine learning classifiers [10]. Naive Bayesian classifiers employ the lowest classification error probability or cost-related average risk. There are studies on cloud-based network intrusion detection analysis; however, deep learning applications for multimedia intrusion detection are uncommon online. Technology in multimedia platforms is the topic. Thus, this research contributes as follows:

- A novel Vision Transformer-based fusion framework that effectively integrates image, video, and text features by patch-wise tokenisation and linear projection into a unified embedding space. In particular, the Feature Interaction Module maps fused visual features into a semantic embedding space aligned with class attribute vectors.
- Multiscale Fusion Module that combines both global and local features from multiple modalities, assigning dynamic attention weights via learnable parameters.
- Developed a graph construction strategy where the fused features are mapped into nodes and connected through edges representing sequential network flow relationships. Employed a multi-head attention mechanism to compute multiple independent self-attention scores for each node.
- Integrated the graph-attention-enhanced node embeddings into a final softmax classification layer, enabling multi-

class intrusion detection for deepfake and spoofing attack categories.

The rest of the article follows. Section 2 explains the literature survey. Section 3 proposes a solution. Section 4 shows comparative research performance and result analysis. Section 5 concludes with suggestions for future work.

II. RELATED WORKS

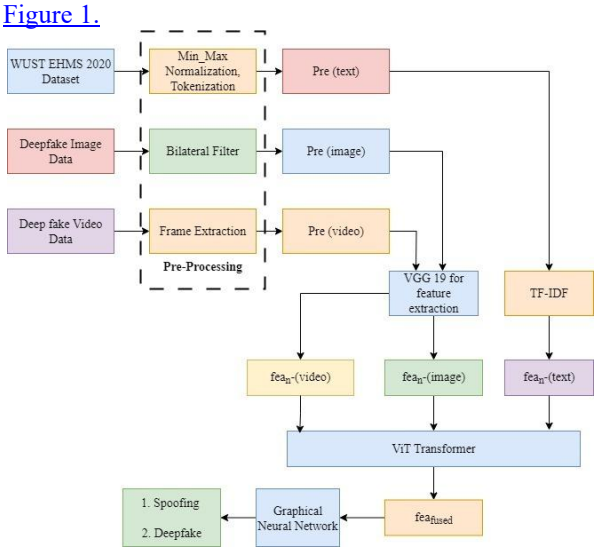
This section summarizes a literature review on neural networks in intrusion detection systems. In [11], a hybrid Hopfield recurrent neural network intrusion detection model was used to produce a hybrid feature selection technique. In [12], a stacking ensemble model is proposed that incorporates a Deep Intrusion Detection System (DIDS), a CNN, and an LSTM. Stacking accuracy improves with the Random Forest meta-classifier. In [13] an authenticated access control mechanism (PA2C) and intelligent journey optimization algorithm-based Radial basis neural network may identify network assaults. In [14], Deep Spoof, a spoofing system that monitors wireless data and forecasts IoMT trends using deep learning, was introduced. In [15], we introduced a Deep Spectral Gated Recurrent Neural Network. Normalize Darknet-IDS data before preprocessing. IDBRA and TFDR identify feature margins. In [16], a CNN with Elephant Herding Optimisation is used to create an effective IoMT IDS. In [17], MADP-IIME employs four machine learning methods to detect malware in IoT-enabled industrial multimedia environments. The deep learning-based hyperparameter search (HPS) convolutional neural network with bi-directional long short-term memory (CBL) detects intrusions [18]. It achieves 99.24% precision, 98.69% recall, 98.97% F-score, and 98.18% accuracy. [19] predicts DDoS attacks using Lyapunov Exponent Analysis and Echo State Networks. Exponential Smoothing forecasts network traffic, and the prediction-error time series is calculated by comparing the forecast to observed traffic. [20] proposes integrating sparsity-denoising operators into CNNs to improve robustness. The preceding description lists several restrictions:

- H2RNN provides strong feature selection but demands high computation, limiting real-time use. The CNN–LSTM stacking ensemble with Random Forest improves accuracy but relies on large labelled datasets and complex training, limiting scalability.
- The PA2C with IntVO-RBNN detects attacks effectively but is sensitive to parameter tuning, making it less adaptable to evolving threats. DeepSpoof identifies spoofing in IoMT but struggles with zero-day attacks because it relies on historical traffic.
- DSGRNN achieves precision, though its heavy preprocessing hinders real-time deployment. The CNN with Elephant Herding Optimisation enhances IoMT detection but increases training time and poses a risk of overfitting on imbalanced data.
- The MADP-IIME framework integrates multiple ML methods but adds deployment complexity and resource demand. The HPS-CBL model achieves high accuracy yet requires intensive computation for hyperparameter search.

These limitations can be addressed by employing multimodal fusion using a Graph Neural Network (MultiMod_GNNnet), thereby improving generalizability across diverse datasets and attack scenarios.

III. PROPOSED METHODOLOGY

The proposed system incorporates multimodal data processing for effective attack detection in healthcare multimedia. The WUSTL EHMS 2020 text dataset is initially preprocessed using Min–Max normalisation to uniformly scale feature values. For the Deepfake Medical Image dataset, image quality is improved by applying a bilateral filter process that retains edges while eliminating noise. The video dataset undergoes frame extraction, transforming temporal sequences into representative keyframes for subsequent processing. Preprocessed text data are converted to numerical form using TF–IDF, and the preprocessed image and video frames are fed into the VGG-19 network to capture high-level visual features. Then, text, image, and video features are combined with a Vision Transformer (ViT) to capture cross-modal contextual relationships effectively. Lastly, the combined multimodal representation is fed into a Graph Neural Network (GNN), which utilises the relational properties of the features to effectively classify malicious and benign attacks in the multimedia healthcare system, as shown in Figure 1.



[Fig.1: Overall Block Diagram Representation of Proposed Attack Detection]

A. Preprocessing of Data

Data normalisation is performed on numerical characteristics before classification or clustering methods that focus on numerical features are applied to text data. We use min-max normalisation, the most common method for normalising data, which transforms feature values to lie within a predetermined range, typically [0–1]. All data relationships are preserved using min-max normalisation. The following equation normalises feature values,

$$v' = \frac{v - \min(A)}{max(A) - \min(A)}\left(new_{max(A)} - new_{min(A)}\right) + new_{min(A)} \quad (1)$$

Where the new normalised value is v'. The original feature value is v. The maximum value for feature A

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

is $w_{max(A)}$. The minimum feature value is A, $min(A)$. While $new_{max(A)}$ and $new_{min(A)}$ indicates the new range's maximum and minimum values. After normalisation, tokenisation was employed to divide the text into words or phrases to enable more thorough and precise processing in subsequent analysis.

Tokenisation is essential for separating text into pieces that may be evaluated individually or in context. Preprocessed text is $pre(text)$. Bilateral filters preprocess picture data. A low-pass Gaussian technique is used to construct the denoising bilateral filter, which incorporates both pixel distance and image intensity. Domain and range filters. Its operation is as follows.

$$m'(a,b) = \sum_{k}\sum_{l} h(a,b;k.l)g(k,l) \qquad (2)$$

a, b; k) a, b close paren and reaction at open paren b close paren toure, $m'(a,b)$ and the reaction at (a,b) o an impulse at (k,l).The definition of $h(a,b;k.l)$ is as follows.

$$h(a_0,b_0,a,b) = \begin{cases} r^{-1}_{t_0,f_0} \exp\left(-\dfrac{(t-t_0)^2+(f-f_0)^2}{2\sigma_d^2}\right)\exp \\ \left(-\dfrac{g(a,b)-g(t_0-f_0)^2}{2\sigma_r^2}; (a,b) \in \delta_{t_0f_0}\right. \\ 0 \quad ; \quad else \end{cases} \qquad (3)$$

In the window, a_0, b_0 is the central pixel, $\delta_{t_0f_0} = \{(a,b): (a,b) \in (a_0-M, a_0+M) \times (b_0-N, b_0+N)\}$; σ_d and σ_r indicate the domain and range Gaussian filters, and r_{a_0,b_0} maintain the average image intensity. Preprocessed images are pre(image). After grayscaling, the set of video frames is indicated as $\vartheta' = \{s'_1, s'_2, \dots s'_n\}$ in video files. $D(j)$ is the frame-to-frame difference between frame j and $j-1$, as illustrated in equation (4),

$$D(j) = \frac{\sum_{m=1}^{w}\sum_{j=1}^{h}\|s'_i(j,m) - s'_{i-1}(j,m)\|}{w \times h}, i \in \{2,3,\dots N\}$$

Frames with a difference greater than the threshold are keyframes. This differential frame selection technique focuses on the maximum value. For instance, $D(j) \geq \frac{D(j-1)D(j+1)}{2}$ indicates that s_i is a differential frame due to its significant difference from other frames in $\{s_{i-2}, s_{i-1}, s_i, s_{i+1}\}$. The differential frames are represented by $\forall_d = \{s_1, s_2, \dots s_U\}$. We filter redundant information by setting a threshold. $\propto$. If $U \leq \alpha$, the algorithm reports the result. Otherwise, it repeats the differential frame extraction procedure on $\forall_d$. The preprocessed videos are $pre(video)$.

Algorithm-1 for preprocessing

Input: raw text (T), raw image (I), raw video(V)
Output: $pre(text), pre(image), pre(video)$
preprocess_text (T):
For each feature column in T that is numerical:
For each value v in feature A
Compute v'
End for
End for
Return $pre(text) = tokenized\_text$
preprocess_image(I):
for each image in I
For each pixel position (a,b)
Apply bilateral filter
End for
End for
Return $pre(image) = filtered\_image$
preprocess_video(V):
differential_frames set
for each video in V
Frames = Extract_Frames(video)
Gray_Frames = Convert_To_Grayscale(Frames)
Keyframes = []
Add frame i to Keyframes
End for
Return $pre(video) = Differential\_Frames\_Set$

B. Feature Extraction by VGG-19 and Term Frequency Inverse Document Method

VGG-19, a deep CNN model trained on ImageNet, extracts feature from preprocessed $pre(img)$ and $pre(video)$ images and videos. VGG19 contains 19 layers and 19 learnable weights for TL paired with FC layers and an output layer. The first layer input size for VGG19 is 224x224x3. Convolutional layer 1 contains 3x2x3x64 weights and 1x1x64 bias. The first layer has 1792 learnable weights and the second 36292. Features retrieved by the first layer are:

$$fea_{out(i)} = A_i^{pre(img)} + \sum_{k=1}^{N-1} \varphi_{i,k}^{N} \times g_k^{N-1} \qquad (4)$$

Where $fea_{out(i)}$ is the output layer and $A_i^{pre(img)}$ is the input image's base value. $\varphi_{i,k}^{N}$ measure the kth feature value, and N-1 displays the result. The first FC layer has 4096 x 25 learnable weights and 1 bias. Between FC layers, dropout is 50%. Learnable final layer weights are 1000 x 4096. Activation yielded feature maps of 1 × 1 × 1000 and 1 × 1 × 4096 for FC1 and FC2.

As a result, the extracted features from images are denoted as $fea_n(img) = fea_{1\_img}, fea_{2\_img}, \dots fea_{n\_img}$. As a result, the extracted features from frames are denoted as $fea_n(video) = fea_{1\_video}, fea_{2\_video}, \dots fea_{n\_video}$. The $pre(text)$ underwent Term Frequency-Inverse Document Frequency (TF-ID), which is calculated using equations (5) to (7)

$$TF(t,a) = \frac{fea(t,a)}{fea(w,a)} \qquad (5)$$

$$IDF(t,a) = ln\frac{N}{\sum(a\epsilon A: t \in a)+1} \qquad (6)$$

$$TF - IDF(t,a,A) = TF(t,a) \times IDF(t,a) \qquad (7)$$

Where, t is a term, a is an attribute, w is the frequency of the term in the attribute, A is a corpus, and N is the number of attributes in the corpus. TfIdf score can effectively highlight terms that are important and relevant to a particular attribute, while filtering out terms that are common across all attributes. As a result, the extracted features from images are denoted as $fea_n(text) = fea_{1\_text}, fea_{2\_text}, \dots fea_{n\_text}$.

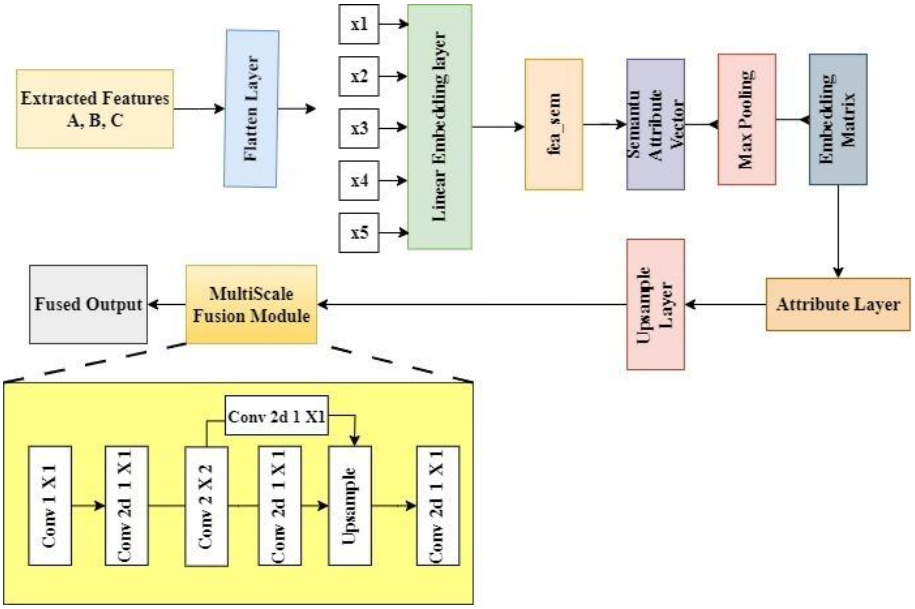
C. Feature Fusion: ViT Transformer

The extracted features, such as

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

$fea_n(img) denoted\ as\ A, fea_n(video)\ denoted\ as\ B$ and as $fea_n(text) denoted\ as\ C$ have to be fused to reduce the dimension, as shown in Figure 2. ViT divides $IN \in R^{A\times B\times C}$ into N fixed-size patches $(P \times P)$ and flattens them into vectors. To embed vectors in a D-dimensional space, a trainable linear projection is used $E \in R^{(P^2,C)\times D}$,

$$flatten(x[i])E\epsilon R^D, i, 1,2, \dots N \qquad (8)$$



[Fig.2: Architecture of Proposed ViT Transformer for Fusion Process]

i. Feature Interaction Module:

We include the semantic feature fea_sem to improve the model's classification once the flatten layer acquires the final feature fusion FU. The feature fusion is projected to the semantic embedding space after generation. The semantic attribute vector $D_\alpha = (u\beta)_{\beta=1}^{\alpha}$ acts as a support vector for the visual feature vis_{fea}, supplementing it with semantic attribute information for alignment. The fusion of features vis_{fea} is transferred to a semantic embedding space that matches the semantic attribute information D_α using the mapping function map. This mapping procedure aligns feature representations with class semantic attributes and allows the model to accurately classify them in a high-dimensional semantic space by transforming the embedding matrix. wei. The attribute score $att(xi)$ measures the confidence level for each attribute in image xi, enabling attribute-class inference. The model uses observed class-attribute information to derive unseen class attributes by manipulating features within the semantic embedding space.

$$att(xi) = map(fea)u\beta, j = \sum_j u\beta, j\left(wei_{sem\_sp}\right)_j \qquad (9)$$

Where the embedding matrix wei maps sem_sp to semantic attribute space. $u\beta, j$ represents the j th component of the vector $u\beta$. The $att(xi)[\propto]$ score indicates the confidence level for the existence of the $\propto$th attribute in an image, video frame, or text $A + B + C$.

ii. Multiscale Fusion Module:

This module integrates both global and local features, with attention weights calculated as follows:

$$\delta_i = \sigma(wei_t.fea(A)_t + fea(B)_t + fea(C)_t + bias_i \qquad (10)$$

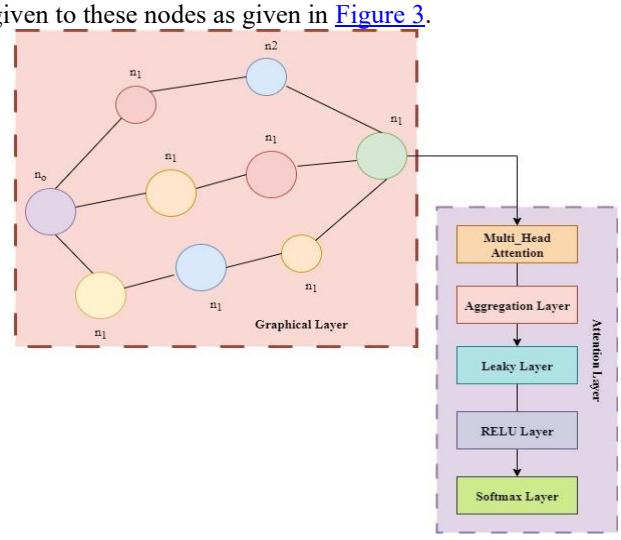
Where, wei_t and $bias_i$ are parameters to be learned, whereas σ denotes a nonlinear activation function. To improve the fusion process, a gated method selectively improves crucial traits while suppressing less significant ones. The ultimate fused representation is obtained as follows.

$$Fea_{fused} = \sum_{t=1}^{n} \propto_t * fea(A + B + C)_t \qquad (11)$$

Where, $\propto_t$ signifies the gating weights assigned to each feature map, and * indicates element-wise multiplication.

D. Graphical Neural Network Based Attack Detection

Initially, the graph is constructed, which comprises nodes. $N = [n0, n1, n2, , , , nN]^T$, wherein the fused features are given to these nodes as given in Figure 3.



[Fig.3: Graphical Neural Network with Attention Layer]

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

$, f\ e\ end\ base, s\ ub\ cap\ V\ an\ subscript\ base, f\ e\ end\ ba\ sub\ e\ fea_V$

an fea_e represent node and edge feature matrices, respectively, to define the properties of the graph. The feU×V is expressed as a feature matrix fea_{mat} s U×V, expressed as $fea_V \in R^{U \times V}$, where V repres, f, e a., a,m, t, open bracket k close bracket represens the edge properties of consecutive network flows and has dimensions × rof $k \times r$. The features of edge $edge_{ij}$ are grouped into the matrix $fea_{mat_{edgeij} \in R^{1,K}}$. The N×N matrix A represents the adjacency of a multigraph. The matrix entries show edge frequency over time. The following requirements apply to A a_{ij}, the element in the i-th row and j-th column: If $a_{ij} = 0$, then there is no edge between nodes v_i and v_j at any time step, i.e., 〖edge〗 _ij∈ 〖fea〗 _mat for all $r \in [1, R_{ij}]^T$, Neigh_(v_i) is the collection of nodes linked to v_i by any of its edges, written as $Neigh_{v_i} = \{v_j | edge_{ij} \in \varepsilon_T\}$, where $r \epsilon [1, R_{ij}]^T$. The leaky layer of $edge_{ij}$ is indicated as

$$edge_{ij} = LeakyReLU(g^T [Wh'_i . Wh'_j] \quad (12)$$

Where, $h_i \epsilon R^f$ is the f-dimensional feature vector of node i, and $W \in R^{f' \times f}$ is a shared weight matrix that linearly translates features from f to f' dimensions. The parameter vector of a feedforward neural network is $g \epsilon R^{2f'}$. The LeakyReLU activation function adds nonlinearity. Normalise the unnormalized attention coefficients. $edge_{ij}$ Using the softmax function for node status updates, as indicated in Equation (13),

$$att_{ij} = softmax_j(edge_{ij}) = \frac{\exp(edge_{ij})}{\sum_{k \in N_i} \exp(edge_{ij})} \quad (13)$$

The Normalized attention coefficients att_{ij} modify node states to act as weights. The hidden representation hid'_i of node i is updated by aggregating information from surrounding nodes using weights. This work adds multi-head attention to increase the model's expressiveness and the stability of self-attention learning. This method generates nodes' self-attention scores using y attention heads and averages the output vectors to improve model generalisation, as shown in Equation (14)

$$hid'_i = \sigma\left(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in N_i} att_{ij}^k W_k hid_j\right)$$

The final output layer is denoted as

$$y' = softmax(W_{out} hid_v^L + b)$$

Where, $hid_v^L \in R^d$ is the final embedding after L number of GNN layers, W_{out} is the weighted matrix which contains intrusion classes, b is the bias. As a result, the final result is classified as a deepfake and spoofing attack.

IV. PERFORMANCE ANALYSIS

A. Experimental Setup

The Learning rate is 0.01, and Adam schedules it. Training uses 100 epochs and 32 batches. Texperiment utilises the NVIDIA GeForce RTX 4090 GPU and employs the PyTorch 1.1.0 deep learning framework.

B. Dataset Description

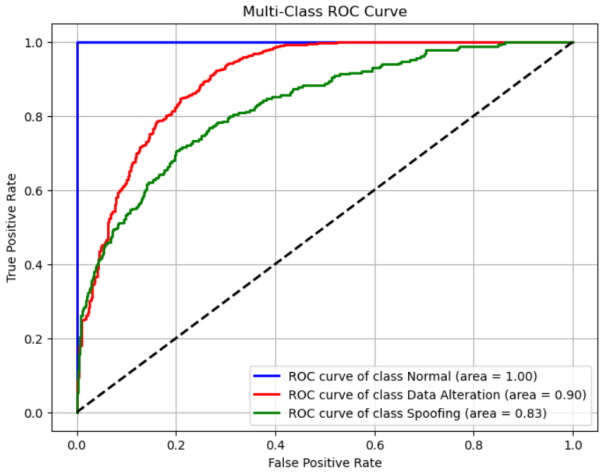
Here, two types of datasets are utilised: 1) WUSTL EHMS 2020 as text data and 2) Image data.

i. WUSTL EHMS 2020 [21] - ARGUS saved network flow traffic and patient biometric data in a "CSV" file. Table 1 displays WUSTL-EHMS-2020 data. It has 35 network flow measures, 8 patient biometrics, and 1 label. We used the Source MAC address tool to assign samples a value of 1 for the attacker's laptop and 0 for the others. This dataset includes man-in-the-middle attacks such as spoofing and data injection.

Table 1: Dataset Description - WUSTL EHMS 2020

Measurement	Value
Data size	4.4mb
No. of normal samples	14,272
No. of attack samples	2046
Total number of samples	16318

ii. Deepfake Image Dataset [22]: It consists of 7535 images in total, whereas 6594 are train images, 629 images are the validation set, and 312 images are the test set. The input images were first auto-oriented to ensure correct alignment based on embedded metadata, then resized to a fixed resolution of 416×416 pixels using a stretch operation to standardise dimensions across the dataset. To enhance feature visibility, contrast stretching was applied, redistributing pixel intensity values to span the full available range.

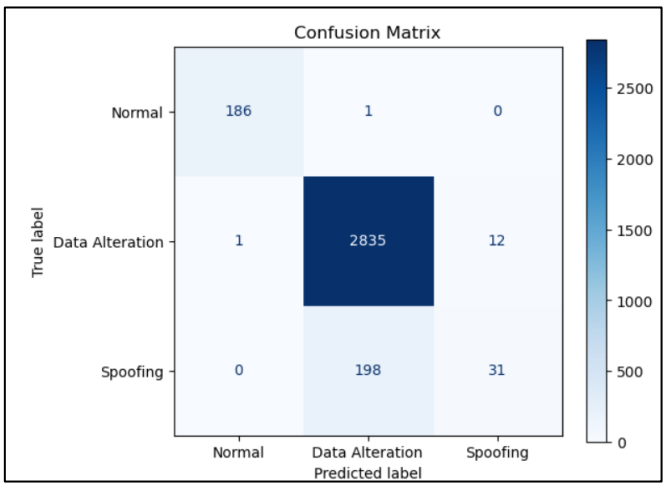


[Fig.4: ROC Curve Using Text Dataset]

Figure 4 shows the ROC curve analysis of the WUSTL-EHMS-2020 dataset across three classes: Normal, Data Alteration, and Spoofing. The blue curve achieves an AUC

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

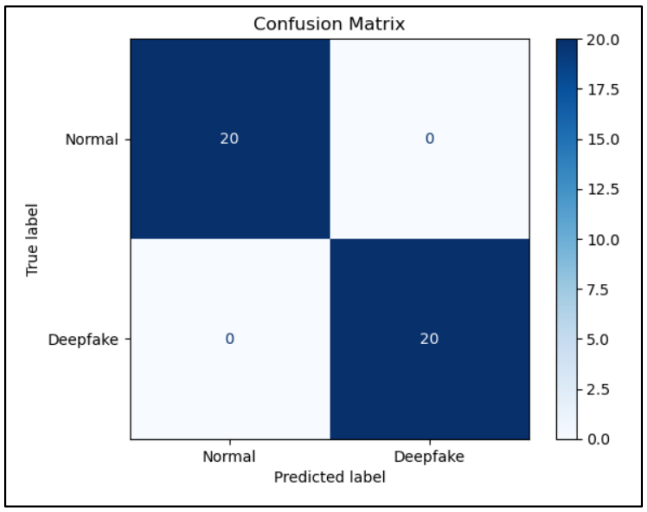
of 1.00, the Data Alteration class achieves an AUC of 0.00, and the red curve represents the Data Alteration class, yielding an AUC of 0.90. The baseline is 0.5; AUC values indicate improved attack classification.



[Fig.5: Confusion Matrix - WUSTL EHMS 2020 Dataset]

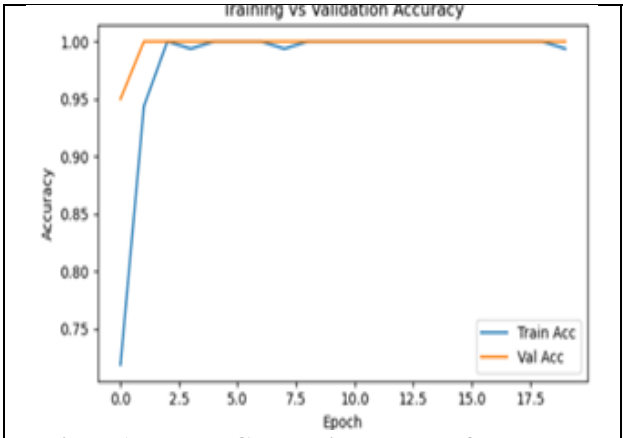
Figure 5 shows the confusion matrix for the proposed ViTfuse_GNN's classification performance using the WUSTL EHMS 2020 dataset. This model indicates the Normal class, correctly identifying 186 instances with only 1 misclassified as Data Alteration and none misclassified as Spoofing. For the Data Alteration class, the model achieves high precision and recall, correctly classifying 2835 instances, with minimal confusion: 1 instance misclassified as Normal and 12 as Spoofing.

However, the Spoofing class shows a comparatively higher misclassification rate: only 31 instances are correctly predicted, while 198 are incorrectly labelled as Data Alteration.

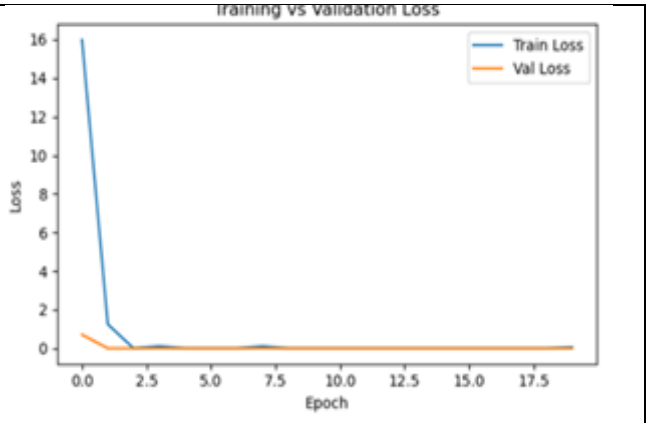


[Fig.6: Confusion Matrix on Deepfake Image Dataset]

The confusion matrix illustrates the classification performance of the proposed ViTfuse_GNN using the deepfake image dataset, as shown in Figure 6. The matrix indicates that all 20 “Normal” instances were correctly classified as “Normal,” and all 20 “Deepfake” instances were correctly classified as “Deepfake,” resulting in zero misclassifications. This yields a perfect classification performance with the best accuracy.

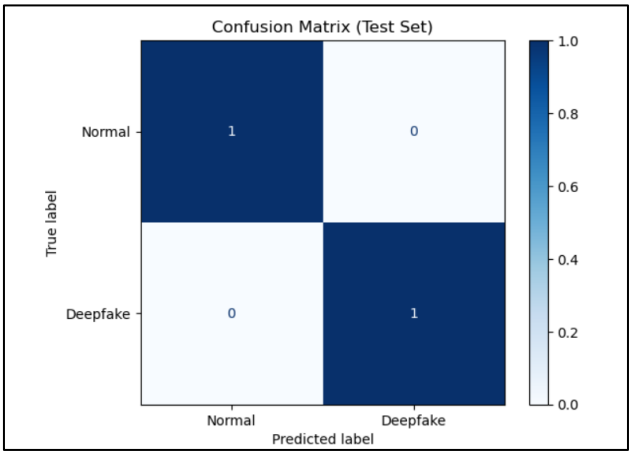


[Fig.7: Accuracy Calculation on Deepfake Image Dataset]



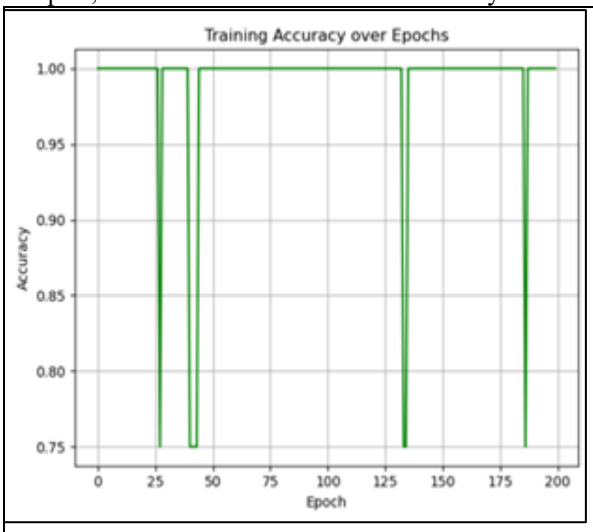
[Fig.8: Loss Calibration on Deepfake Image Dataset]

Fig.7: shows the accuracy plot for the proposed Vithusan method on the deepfake image dataset. Training accuracy starts at around 72%, quickly increases to over 95% by the second epoch, and remains close to perfect thereafter. Validation accuracy also shoots up sharply, crossing almost 100% at the second epoch and remaining consistent across the rest of the training iterations. Figure 8 shows the loss calculation. By epoch 1, this drops significantly to approximately 1.2, and by epoch 2, it is nearly 0.05, remaining at this near-zero level for the rest of the training course. The minimal variation of training and validation loss across all epochs signals robust generalisation without overfitting.

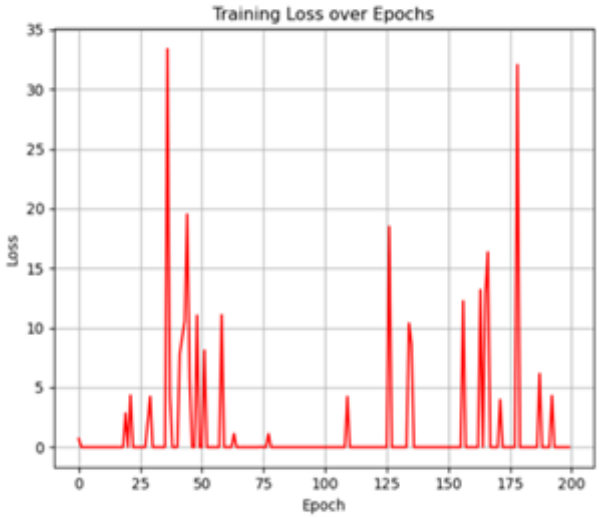


[Fig.9: Confusion Matrix on Deepfake Video Dataset]

The confusion matrix illustrating the classification performance of the proposed ViTfuse_GNN using the deepfake video dataset is shown in Figure 9. Out of the two test samples, one "Normal" instance was correctly identified as "Normal", and one "Deepfake" instance was correctly classified as "Deepfake". There are no false positives or false negatives, resulting in the best accuracy.



[Fig.10: Accuracy Calculation on Deepfake Video Dataset]



[Fig.11: Loss Calibration on Deepfake Video Dataset]

Figure 10 shows the accuracy plot of the suggested ViTfuse_GNN approach utilising the deepfake video dataset. Here, the x-axis and y-axis represent epochs and accuracy values. For epochs like 25, 50, 75, and 100, the accuracy is approximately 1, i.e., 100%. Accuracy transiently falls to about 0.75 before bouncing back in further epochs. From Figure 11, we can see oscillations that likely correspond to the same epochs when training accuracy briefly fell, indicating instances of instability during optimisation.

Table 2: Spoofing Attack Results

Number of Patients	With Attack or Without Attack	Accuracy	Precision	Recall	F1-Score
Paitent-1 Data	Before Attack	0.99	0.99	0.99	0.99
	Spoofing Attack	0.94	0.93	1.00	0.96
Patient-2 Data	Before Attack	0.98	0.72	0.14	0.23
	Spoofing Attack	0.93	0.88	0.71	0.94
Patient-3 Data	Before Attack	0.96	0.92	0.94	0.73
	Spoofing Attack	0.91	0.94	0.92	0.94

Table 3: Deepfake (Image) Results

Number of Patients	With Attack or Without Attack	Accuracy	Precision	Recall	F1-Score
Paitent-1 data	Before attack	0.99	0.99	0.99	0.99
	Deepfake attack	0.99	0.99	0.99	0.99
Patient-2 data	Before attack	0.99	0.99	0.99	0.99
	Deepfake attack	0.99	0.99	0.99	0.99
Patient-3 data	Before attack	0.99	0.99	0.99	0.99
	Deepfake attack	0.99	0.99	0.99	0.99

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Table 4: Deepfake (Video) Results

Number of Patients	With Attack or Without Attack	Accuracy	Precision	Recall	F1-Score
Paitent-1 data	Before attack	0.99	0.99	0.99	0.99
	Deepfake attack	0.99	0.99	0.99	0.99
Patient-2 data	Before attack	0.99	0.99	0.99	0.99
	Deepfake attack	0.99	0.99	0.99	0.99
Patient-3 data	Before attack	0.99	0.99	0.99	0.99
	Deepfake attack	0.99	0.99	0.99	0.99

Table 5: Overall Comparative Analysis

Parameters	ViTfuse_GNN
Accuracy (%)	0.995
Precision (%)	0.987
Recall (%)	0.994
F1-score (%)	0.972

V. CONCLUSION

The security and privacy of multimedia data should be taken seriously in light of today's rapidly advancing multimedia technologies, especially in applications that handle sensitive data. This research conducted an intrusion detection investigation to assess the false alarm rate using the ViTfuse_GNN architecture, which is appropriate for smart healthcare applications. Future work focuses on incorporating adversarial training and defence mechanisms to enhance resilience against sophisticated cyberattacks and adversarial perturbations targeting multimedia healthcare content.

DECLARATION STATEMENT

Some of the cited references are older and are noted explicitly as [6], [8], and [9]. However, these works remain significant for the current study, as they are pioneering in their fields.After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been sponsored or funded by any organization or agency. The independence of this research is crucial to affirming its impartiality, as it has been conducted without external influence.
- **Ethical Approval and Consent to Participate:** The data provided in this article are exempt from the requirement for ethical approval or participant consent.
- **Data Access Statement and Material Availability:** The data resources of this article are publicly accessible.
- **Author's Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCE

1. Mubashar A, Asghar K, Javed AR, Rizwan M, Srivastava G, Gadekallu TR, et al. Storage and proximity management for centralised personal health records using an IPFS-based optimisation algorithm. J Circ Syst Comp. (2021) 15:2250010. doi: 10.1142/S0218126622500104
2. Iwendi C, Khan S, Anajemba JH, Mittal M, Alenezi M, Alazab M. The use of ensemble models for multi-class and binary classification to improve intrusion detection systems. Sensors. (2020) 20:2559. doi: 10.3390/s20092559
3. Yeng PK, Nweke LO, Woldaregay AZ, Yang B, Snekkenes EA. Data-driven and artificial intelligence (AI) approach for modelling and analysing healthcare security practice: a systematic review. In: Arai K, Kapoor S, Bhatia R, editors. Intelligent Systems and Applications. Cham: Springer (2021). doi: 10.1007/978-3-030-55180-3_1
4. Subasi A, Algebsani S, Alghamdi W, Kremic E, Almaasrani J, Abdulaziz N. Intrusion detection in smart healthcare using bagging ensemble classifier. In: International Conference on Medical and Biological Engineering. Cham: Springer (2021) p. 164–71. doi: 10.1007/978-3-030-73909-6_18
5. Ashok Kumar D, Venugopalan SR (2018) A novel algorithm for network anomaly detection using adaptive machine learning. In: Progress in Advanced Computing and Intelligent Engineering. Springer, pp 59–69. DOI: https://doi.org/10.1007/978-981-10-6875-1_7
6. Deebak BD, Muthaiah R, Thenmozhi K, Swaminathan PI. IP Multimedia Subsystem—an intrusion detection system. SmartCR. 2013;3(1):1–13. doi: 10.6029/smartcr.2013.01.001.
7. *Deng X, Li Y, Weng J, Zhang J. Feature selection for text classification: A review. Multimed Tools Appl. 2019;78(3):3797–3816. doi:* 10.1007/s11042-018-6083-5.
8. Cortes C, Vapnik V. Support-vector networks. Mach Learn. 1995;20(3):273–297. DOI: https://doi.org/10.1007/BF00994018
9. Cristianini N, Shawe-Taylor J, et al. (2000) An introduction to support vector machines and other kernel-based learning methods, Cambridge University Press, Cambridge. DOI: https://doi.org/10.1017/CBO9780511801389
10. Deng X, Li Y, Weng J, Zhang J. Feature selection for text classification: A review. Multimed Tools Appl. 2019;78(3):3797–3816. doi: 10.1007/s11042-018-6083-5.
11. Parimala, R., & Gunasekaran, S. (2025). H2RNN: an automatic intrusion detection model for a cloud environment using a hybrid feature selection model and a hybrid Hopfield recurrent neural network—Journal *of Computer Virology and Hacking Techniques*, *21*(1), 22. DOI: http://doi.org/10.1007/s11416-025-00566-0
12. Vishwakarma, M., & Kesswani, N. (2025). StaEn-IDS: An Explainable Stacking Ensemble Deep Neural Network-based Intrusion Detection System for IoT. *IEEE Access*. DOI: https://doi.org/10.1109/ACCESS.2025.3582391
13. Kumar, A., Batta, P., Rathore, P. S., & Ahuja, S. (2025). Secure healthcare data-sharing and attack-detection framework using a radial basis neural network. *Scientific Reports*, *15*(1), 15432. DOI: https://doi.org/10.1038/s41598-025-99676-4
14. Gao, D., Ou, L., Liu, Y., Yang, Q., & Wang, H. (2024). DeepSpoof: Deep reinforcement learning-based spoofing attack in cross-technology multimedia communication. *IEEE Transactions on Multimedia*, *26*, 10879-10891. DOI: https://doi.org/10.1109/TMM.2024.3414660
15. Dhiyanesh, B., Asha, A., Kıruthıga, G., & Radha, R. (2024). Enhancing Healthcare Data Security in Fog Computing: A Deep Spectral Gated Recurrent Neural Network-Based Intrusion Detection System Approach. DOI: https://doi.org/10.21203/rs.3.rs-3970408/v1
16. Praveena Anjelin, D., & Ganesh Kumar, S. (2024). An effective classification using enhanced elephant herding optimisation with a convolutional neural network for intrusion detection in an IoMT architecture. *Cluster Computing*, *27*(9), 12341-12359. DOI: https://doi.org/10.1007/s10586-024-04512-5
17. Pundir, S., Obaidat, M. S., Wazid, M., Das, A. K., Singh, D. P., & Rodrigues, J. J. (2023). MADP-IIME: malware attack detection protocol in IoT-enabled industrial multimedia environment using machine learning approach. *Multimedia Systems*, *29*(3), 1785-1797. DOI: https://doi.org/10.1007/s00530-020-00743-9
18. Pustokhina, I. V., Pustokhin, D. A., Lydia, E. L., Garg, P., Kadian, A., & Shankar, K. (2022). Hyperparameter search-based convolutional

neural network with Bi-LSTM model for intrusion detection in a multimedia big data environment. *Multimedia Tools and Applications*, *81*(24), 34951-34968. DOI: https://doi.org/10.1007/s11042-021-11271-7

19. Salemi, H., Rostami, H., Talatian-Azad, S., & Khosravi, M. R. (2022). LEAESN: Predicting DDoS attacks in healthcare systems using Lyapunov exponent analysis and echo state neural networks. *Multimedia Tools and Applications*, *81*(29), 41455-41476. DOI: https://doi.org/10.1007/s11042-020-10179-y
20. Shi, X., Peng, Y., Chen, Q., Keenan, T., Thavikulwat, A. T., Lee, S., ... & Lu, Z. (2022). Robust convolutional neural networks against adversarial attacks on medical images. *Pattern Recognition*, *132*, 108923. DOI: https://doi.org/10.1016/j.patcog.2022.108923
21. A. Hady, A. Ghubaish, T. Salman, D. Unal and R. Jain, "Intrusion Detection System for Healthcare Systems Using Medical and Network Data: A Comparison Study," in IEEE Access, vol. 8, pp. 106576-106584, 2020. DOI: 10.1109/ACCESS.2020.3000421. https://ieeexplore.ieee.org/document/9109651
22. https://universe.roboflow.com/medical-deepfake-image-dataset/medical-deepfake-image-dataset/dataset/1