SCIENTIFIC CULTURE
www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 11, No. 3.2, (2025), pp. 1600-1613
Open Access. Online & Print

DOI: 10.5281/zenodo. 113225122

ENHANCED IOMT SECURITY AND PREDICTIVE HEALTHCARE SERVICES BY HYPERPARAMETER-OPTIMIZED DEEP LEARNING MODELS

Veena R S[1*], Nandini Prasad K S[2], Mesith Chaimanee[3], Mithileysh Sathiyanarayanan[4], Asokan Vasudevan[5], Divyashree H B[6], Nasyra Ab. Jamil[7]

[1] Department of Information Science and Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore, India. veenars1975@gmail.com; Orcid: https://orcid.org/0000-0003-1368-0634
[2] Department of Information Science and Engineering (ISE), Dayananda Sagar Academy of Technology and Management, Bangalore, India. drnandini.prasad1@gmail.com; Orcid: 0000-0002-9578-8085
[3] Shinawatra University, Thailand. mesith.c@siu.ac.th
[4] Scientist, MIT Square Services Private Limited, London, UK. s.mithileysh@gmail.com
[5]Asokan Vasudevan- Faculty of Business and Communications, INTI International University, Persiaran Perdana BBN Putra Nilai, 71800 Nilai, Negeri Sembilan, Malaysia. Email: asokan.vasudevan@newinti.edu.my; Orcid: https://orcid.org/0000-0002-9866-4045
[6]Research Fellow, Wekerle Business School, Budapest, Jázmin u. 10, 1083 Hungary.
[7]Department of Electronics and communication Engineering, Dayananda Sagar University, India. divyabalachandra94@gmail.com
[8] Senior Lecturer, Nasyra Ab. Jamil, Faculty of Business and Communications, INTI International University, Negeri Sembilan, Malaysia. nasyra.jamil@newinti.edu.my; https://orcid.org/0009-0005-9792-6540

Received: 27/07/2025
Accepted: 27/08/2025

ABSTRACT

The Internet of Medical Things (IoMT) is a subset of the IoT that enables healthcare providers to provide patients with more tailored and hassle-free electronic health records (e-health). It is slowly changing the healthcare business by using its capabilities to facilitate the smooth transfer of medical data, improving tailored healthcare services. With IoMT, medical services can be delivered more quickly and at a lower cost, and emergency responses can be more efficiently coordinated. The fact that malicious actors with ill intentions can infect these network systems raises privacy and security concerns. IoMT vulnerabilities pose a threat to patients' privacy and health records. The study advocated employing hyperparameter-optimized Deep Learning representations to address these challenges and construct more resilient security solutions. One major challenge for AI models is extracting valuable data from these reports. Decisions about patients' health can be better informed by overcoming these obstacles, especially when it comes to extracting critical elements, including symptoms, comparison/priors, technique, finding, and perception. Researchers in the study suggested using a combination of pre-trained deep learning algorithms and features extracted from pre-processed data to make disease predictions. A competitive swarm optimizer with mutated agents (CSO-MA) is used to fine-tune the model. The attacks are finally recognized with a multi-head attention method (BDMLSTM) and a modified Mogrifier extended short-term memory model. The Election Optimizer Algorithm (EOA) fine-tunes the BDMLSTM parameters to increase the classification accuracy. Using the ECU-IoHT and WUSTL-EHMS-2020 datasets as examples, the experimental findings showed that the suggested model outperformed the current best-practice models.

KEYWORDS: Internet of Medical Things; Election Optimizer Algorithm; Security; Competitive Swarm Optimizer Mutated Agents; Process Innovation.

1. INTRODUCTION

1.1. Background on IoMT and Cybersecurity

The Internet of Medical Things (IoMT), a specialized domain within the Internet of Things (IoT), plays a pivotal role in enabling intelligent healthcare infrastructure. Through the integration of medical devices, wearable sensors, and communication technologies, IoMT facilitates real-time patient monitoring and dynamic management of medical records [1]. Applications of IoMT span across critical areas such as remote diagnostics, emergency alerts, and personalized treatment planning [2]. Despite these benefits, the use of wireless communication and open Internet connectivity exposes IoMT systems to numerous security threats. Cyber intrusions including Denial-of-Service (DoS) attacks, malware propagation, unauthorized access, and data manipulation jeopardize both patient safety and data confidentiality [3]. As healthcare environments adopt more connected systems, ensuring robust cybersecurity in IoMT networks has become an essential priority.

1.2. Need for Intrusion Detection Systems (IDS) in IoMT

Given the increasing sophistication of cyberattacks, conventional security mechanisms are insufficient to defend dynamic IoMT environments [4]. Intrusion Detection Systems (IDS) are crucial components designed to detect anomalies or unauthorized access attempts in network traffic. In IoMT, IDSs must handle high-dimensional, real-time data from heterogeneous sources such as ECG monitors, glucometers, and blood pressure sensors [5]. These systems are often embedded within WSNs, further complicating the security landscape due to their decentralized structure and energy constraints. An efficient IDS must therefore be capable of learning from streaming health data and distinguishing between benign anomalies and genuine threats [6].

1.3. Deep Learning for Medical Intrusion Detection

Deep Learning (DL) has demonstrated significant advantages over traditional methods in terms of feature extraction and adaptability [7]. Particularly in the realm of IoMT, DL models such as Convolutional Neural Networks (CNNs), Long Short-Term Memory networks (LSTMs), and hybrid architectures offer scalable solutions for detecting complex attack patterns [8]. Unlike rule-based systems, DL algorithms autonomously learn hierarchical features from raw inputs, enabling better generalization to novel threats. Moreover, combining different model types—such as CNNs for spatial features and LSTMs for temporal patterns—enhances detection accuracy [9]. This study proposes the integration of pre-trained DL models with custom enhancements to address both disease prediction and anomaly detection in IoMT [10].

1.4. Research Challenges and Objectives

Despite the promise of DL models, several challenges persist. These include the lack of contextual awareness in sequential models, inadequate representation of medical terminologies in standard embeddings, and difficulty in handling high-dimensional feature spaces [11]. To address these issues, this research introduces a hybrid architecture leveraging a Bidirectional Mogrifier LSTM (BDMLSTM) augmented with multi-head attention mechanisms. Additionally, swarm-based optimization techniques are employed for hyperparameter tuning to maximize model performance. The study also proposes an end-to-end pipeline for both disease classification using image data and intrusion detection using network telemetry. The goal is to improve detection rates while ensuring model interpretability and computational efficiency.

1.5. Paper Structure

The remainder of paper is organized as follows: Section 2 reviews related research in IoMT security and DL-based intrusion detection. Section 3 details the proposed methodology, including model design and optimization strategies. Section 4 presents experimental results and comparative analyses. Section 5 concludes the study and outlines future directions.

2. RELATED WORKS

Recent research on securing the Internet of Medical Things (IoMT) has emphasized the use of intelligent deep learning frameworks and federated learning-based models for anomaly detection. Zukaib et al. [12] introduced a meta-learning-based IDS, Meta-IDS, that significantly enhances the classification accuracy of known and zero-day threats in IoMT networks using multiple benchmark datasets. Their work highlighted the importance of model adaptability and reduced misclassification rates. Similarly, Alalhareth and Hong [13] proposed a meta-learning-driven ensemble IDS that dynamically selects base classifiers based on input features. Their approach demonstrated superior

performance over conventional ensemble methods in terms of precision, recall, and F1-score. Alzubi et al. [14] presented a hybrid framework combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks to improve detection accuracy in healthcare edge environments. Their architecture successfully captured both spatial and temporal characteristics of IoMT traffic. Arif, F., et al. [15] propose an intelligent model based on software defined networking and deep learning (DL) to handle the heterogeneous, complex, and distributed architecture. To tackle the imbalance challenge, the proposed model utilizes generative adversarial network (GAN) to generate plausible synthetic data for minor class traffic. It combines autoencoder-driven DL models with reconstruction error and Wasserstein distance-based GAN. When compared to naive and advanced techniques, the proposed model produced noticeably better results on an imbalance dataset and outperformed these techniques by 4.78% and 4.54% in terms of accuracy and F1-score values, correspondingly. Sun et al. [16] integrated Particle Swarm Optimization with AdaBoost classifiers to develop an effective IDS for smart healthcare. Their model selected optimal features for classification, achieving a recall rate above 96% on NSL-KDD. Praveena Anjelin and Ganesh Kumar [17] applied Enhanced Elephant Herding Optimization with CNNs, achieving improved intrusion detection while reducing false positives. Their work also focused on hyperparameter tuning to increase real-time responsiveness. Shambharkar and Sharma [18] proposed deep learning-empowered CatEmb models that leverage patient flow data for intrusion prediction, surpassing existing approaches such as LinSVM and ConvSVM in detection rate. Zukaib et al. [19] later extended their work by combining federated learning with biased classifiers and clustering techniques in Meta-Fed IDS, a multi-phase system capable of detecting novel threats in distributed IoMT architectures. Franklin and Pranggono [20] developed anomaly-based IDSs using hyperparameter-optimized CNN-LSTM models, validating their work on benchmark datasets with enhanced precision and recall values. They applied SMOTE to address data imbalance and used random search for tuning. Zhang [21] utilized SV-SMOTE for oversampling and XGBoost for intrusion classification. Their method outperformed traditional random forests and provided strong performance in multiclass network attack detection. Kumar et al. [22] advanced the domain with a CNN-BLSTM-GRU hybrid deep learning model, demonstrating 99.25% accuracy in botnet detection within IoMT environments. Dhanushkodi et al. [23] introduced an optimal PLSTM classifier enhanced with attention mechanisms and hyperparameter tuning using the OOYO algorithm. Their approach showed robust generalization across datasets like N-BaIoT and IoT-Healthcare. In summary, while existing works offer various architectural innovations, limitations remain in balancing class distributions, ensuring cross-device generalizability, and providing explainable outcomes. This study builds upon these insights by proposing a BDMLSTM-based hybrid model fine-tuned via swarm and election-inspired optimization strategies.

3. PROPOSED METHODOLOGY

The several cyber-attacks that were taken into account by the projected system are detailed in this section. Here, using Figure 1, we explain the idea of the deep neural system and the approach of the suggested organization.

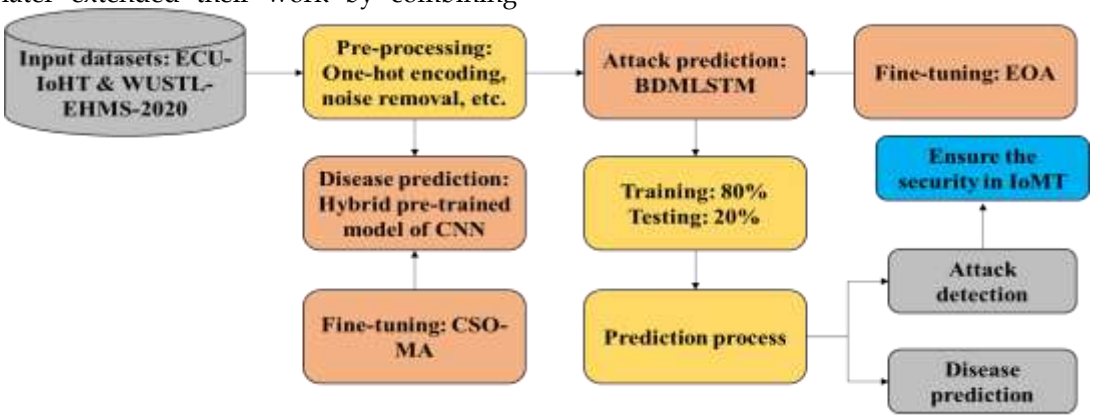

Figure 1: Workflow of Research Work.

3.1. Anomaly Detection

An important step in cyber safety analysis is

anomaly detection, which seeks out out-of-the-ordinary data points in a dataset. According to their characteristics, anomalies can be categorized as either point, collective, or contextual. The immersion of massive amounts of data in an IoT context can give rise to new anomalies or alter existing ones. Attacks like ARP spoofing, denial of service, Nmap PortScan, and Smurf are all taken into account by the suggested model. To avoid IP conflicts in networks, Address Resolution Protocol (ARP) offers the connotation between IP addresses besides MAC addresses. Another name for ARP spoofing is ARP poisoning; in this attack, the attackers receive data flow from the imagined host and send it over the local area network along with bogus data. A permitted resource cannot be accessed in a DoS attack. The scanning tool that may detect Nmap scanning activity in order to find vulnerabilities is Network Mapper (Nmap). One sort of distributed denial-of-service assault is the smurf attack, which mimics the behavior of ping floods while taking use of transmission networks' quirks to significantly increase the capacity of attack traffic.

3.2. Description of Dataset

While developing and testing network intrusion detection systems, numerous researchers made use of various datasets, including DARPA, KDD Cup 99, NSL-KDD, Moore, BOT-IoT, ToN-IoT, SCADA. The security of healthcare devices, however, has not been tested using any of the datasets listed above [24]. Consequently, this paper's suggested deep learning approach is evaluated using a unique dataset, ECU-IoHT [25], in the heath care area. There are 111,207 samples in the collection, covering both typical and unusual attacks. Table 1 provides a full explanation of the sum of counts for various attack labels in the original dataset, including normal, ARP Spoofing, Smurf attacks.

Table 1: Explanation of ECU-IoHT Dataset.

Category	ECU-IoHT		ECU-IoHT in Proposed Scheme	
	Counts	Counts	Training (80%)	Training (20%)
Smurf Attack	77,920	77,920	62,218	15,642
ARP Spoofing	2359	2359	18,780	4673
DoS Attack	639	639	525	114
Nmap PortScan	6836	6836	5510	1326
No Attack/Normal	23,453	23,453	18,780	4673

3.2.1. Description of the Dataset

The WUSTL-EHMS-2020 dataset [26] was used in this investigation. It integrates patient biometric network flow metrics. These data come from a real-time EHMS (Enhanced Healthcare Monitoring System) testbed. An infrastructure for networks, a control sensors, and a data-transmitting gateway make up the testbed. Using routing and switching techniques, data is collected from patient-attached sensors, sent gateway, and finally forwarded to a dedicated server for viewing. Patient biometric data and network flow metrics were the primary goals of the EHMS testbed's design. A multi-sensor board, a data detection system (IDS), a dedicated network, a attacker, and a gateway or central control hub are the six essential parts of its system. Medical Expo's PM4100 Six Pe Multi-Sensor Board has four sensors that track vital signs including electrocardiograms (ECGs), heat, blood pressure, and blood oxygen saturation (SpO2). The data is sent to a Windows laptop via a USB port, which acts as the intermediary. In addition to sending the data to a server for processing, the gateway also provides a graphical user interface (GUI) to display the data visually. The data is collected and analyzed by the server, which is running an Ubuntu system. This helps in making educated medical decisions. A router is in charge of allocating dynamic IP addresses, and an Ethernet switch links the server, intrusion detection system, and attack simulator. In order to make crucial decisions regarding traffic packets, the IDS uses Argus, a program that monitors network flows. This software collects biometric data and metrics related to network flows. In order to mimic possible dangers in healthcare monitoring systems, the virtual attacker uses Kali Linux to generate risks like data spoofing or changing patient data while transmission.

3.3. Data Pre-processing

In its original form, the ECU-IoHT dataset has eleven features. By extracting five features—type, source packets, protocol, and length—from this set, we may investigate the suggested system's efficiency. Both numerical and category features have been retrieved. Since only numerical features are taken into account in the experiments, the category features are converted to numerical topographies using encoding, as seen in Algorithm 1. The dataset is represented in Algorithm 1 by D(f1,f2,...,fn),1 < n < N, where N is the total sum of features in dataset.

Algorithm 1: One-hot encoding for encoding unconditional data
Input: $D(f1, f2, \ldots, fn)$
Output: $Dencoded(f1encoded, \ldots, fn\ encoded)$
$For\ i\ from\ 1\ to\ N\ do$
$If\ (fi\ is\ a\ categorical\ input)$
$Encode\ using\ one\ hot\ encode\ method$
$End\ if$
$End\ for$

The dataset (WUSTL-EHMS-2020) was pre-processed multiple times to guarantee relevant and high-quality data before it was used. Feature

selection involved determining which qualities were most important for intrusion detection, normalization involved removing outliers and inconsistencies from the data, and data cleaning involved removing inconsistent or inconsistently distributed values. Improving the dataset for training and performance of the model relied heavily on this pre-processing.

3.4. Disease Prediction using Hybrid Pre-Trained Models

Even though, the main target of the research work is to identify the attacks in IoMT domain, the work focuses on both objectives such as disease prediction and attack detection on IoMT domain.

3.4.1. First model (MobileNet + DenseNet121).

In order to achieve better feature extraction, the suggested hybrid model incorporates transfer learning to merge the capabilities of two models, namely DenseNet121 and MobileNet. Two independent branches make up the model's internal architecture; each of these branches uses a pre-trained model to process input image and extract features. Following a depthwise convolution, the MobileNet model contains 28 convolutional layers. A reduced number of parameters and efficient computation are both made possible by this design. In order to make training faster and more efficient, the input to MobileNet branch is reduced in size from pixels. In contrast, the DenseNet121 model makes use of 121 convolutional layers that are densely connected to one another. With this design, feature reuse and gradient flow are both enhanced, leading to higher accuracy. Resizing the input to the DenseNet121 branch from 50 × 50 pixels to 32 x 32 pixels is also done. The combined capabilities of MobileNet besides DenseNet121, when used in a hybrid perfect, can greatly enhance the model's performance. While DenseNet121's dense connectivity can enhance feature propagation and reuse, MobileNet's depthwise separable convolutions can lessen memory footprint. Dense layers for binary classification utilizing SoftMax activation can be given the combined output from both branches. In sum, with a total of 10,398,578 trainable parameters, this hybrid model can offer a more reliable and precise answer for illness prediction. For classification problems, function is widely employed. The output of each branch from the layer is combined and routed through two dense layers, one with 64 units and the other with 16 units. Using the Adam function, the model is trained.

3.4.2. Second Model (MobileNetV2 + Efficientnetv2_b0).

The hybrid model known as (MobileNetV2 + EfficientNetV2_b0) combines the two pre-trained models, EfficientNetV2_b0 and MobileNetV2, to enhance the precision of disease prediction. There are 53 layers in the MobileNetV2 model, and one of them is a depthwise separable convolution layer, which allows for faster convolutional operations without compromising accuracy. To aid the model in learning more complicated characteristics, it also incorporates inverted residual blocks that make advantage of skip connections. Conversely, the EfficientNetV2_b0 model employs a compound scaling technique to maximize efficiency and scalability. It is able to zero in on the most important aspects of the input picture thanks to its attention mechanism and 24 layers, which include numerous convolutional layers. Both branches' inputs are downsized from 50 x 50 to 32 x 32 pixels, and 8,342,274 parameters are trainable. The hybrid model involves running the image through two separate models: EfficientNetV2_b0 for high-level feature extraction and MobileNetV2 for low-level feature extraction. Combining the two models' outputs, they are fed into two dense layers that use the Softmax function to conduct binary classification. The hybrid model is able to improve the accuracy of disease prediction by capturing a wider range of features, thanks to the use of two pretrained models with different architectures. To improve our deep learning model's training process, we used the CSO-MA optimizer with an adaptive learning rate besides decay The optimizer uses adaptive learning rates to speed up the learning process and increase convergence by dynamically adjusting the learning rate training. Learning rate decay, which involves a slow but steady decrease in the learning rate, was also included. In order to avoid instability and overshooting in later phases of training, this strategy allows the model its biases and weights more efficiently as it tactics convergence.

3.4.3. Fine-Tuning Using CSO-MA

An effective swarm-based technique for solving multiple dimensional optimization problems is the competitive which was proposed by [27]. For instance, we utilized CSO to investigate a power dispatch, which is usually a complicated and to choose variables for models. CSO initially generates a set of potential solutions at random before minimizing a given continuous function f(x) across a compact space ¡ ¬ that the user specifies. With correspondingly random velocities, they resemble a swarm of n particles positioned at x1,• • •,xn. v1,• • • ,vn. When solving design challenges, each particle

represents a potential design; the best design is found when all of the designs converge.

After the primary swarm is produced, at each repetition we indiscriminately rift the swarm into [n/2] pairs values. At repetition t, we classify x_t^i as the winner and xjt as the loser if f(x_t^i)<f(x_j^t).

v_j^(t+1)=R_1⊗v_j^t+R_2⊗(x_i^t-x_j^t)+ϕR_3⊗(x¯^t-x_j^t) (1)

And x_j^(t+1)=x_j^t+v_j^(t+1) (2)

where R_1,R_2,R_3 each consist of a random vector with entries taken at random from the set U(0, 1). This procedure ⊗ characterizes element-wise multiplication besides the vector x¯^t is repetition t. The social factor φ determines how nearby particles affect the loser; a high value aids in increasing swarm diversity (but may affect convergence rate). The procedure continues to iterate until some endpoint is reached. Simulation studies have consistently demonstrated that CSO achieves competitive or better performance than numerous state-of-the-art swarm-based besides evolutionary procedures, including multiple improved variants of PSO. Using a range with dimensions up to 5000, we compared CSO presentation with that of state-of-the-art EAs and discovered that CSO consistently produced the fastest and highest quality results.

A) Competitive swarm optimizer with mutated agents Competitive swarm optimizer with mutated agents (CSO-MA) is an improved version of CSO that was proposed by [28]. The variation selects a loser atom p as an manager and erratically assigns it to a variable indexed as after iteratively coupling up the swarm in pairs. q randomly changes the value of x_pq to either 〖xmax〗_q or 〖xmin〗_q, where 〖xmax〗_q and 〖xmin〗_q represent, respectively, q-th variable. We conduct this particle that is not in charge of the swarm's movement; if the current is already near the global optimum, then this change an opportunity to optimum.

When compared to regular CSO, CSO-MA stands out due to its mutation stage (the purple box). See for more on how the mutation aims to boost solution variety and delay convergence to a best by enabling particles to explore farther portions of the space. Let n be swarm size besides let D dimension of problem. The computational complexity of CSO is O(nD) The Times Novel Roman font φ = 0.3 besides the rest of its limits besides PSO are altogether set to evasion values. We used early stopping to improve our model's generalizability and reduce the likelihood of overfitting. When a model grows too intricate and starts to remember just the training data rather than discovering patterns in the real world, this is called overfitting. Early stopping is a training technique that involves continuously evaluating the model's presentation dataset. The training process is halted prematurely to avoid overfitting after the model's validation set performance begins to decline or reaches a plateau. During validation, we can choose the most effective model to ensure that the final important illustrations from the data with little influence from noise and irrelevant patterns. The mix of an adaptive learning halting makes our deep learning model very robust and effective on disease prediction tasks.

3.5. Attack Detection using Bidirectional Mogrifier LSTM (BDMLSTM)

We offer a tweaked version of the Mogrifier LSTM [48] for IoMT attack detection in this paper. The idea behind the model is to collect info about context that regular LSTM isn't good at extracting. When associated to the LSTM model, the BiLSTM model performs far better since it handles information. Long short-term memory (LSTM) functions include:

I_t=σ(X_t W_xi+H_(t-1) W_hi+b_i) (3)

F_t=σ(X_t W_xf+H_(t-1) W_hf+b_f) (4)

O_t=σ(X_t W_xo+H_(t-1) W_ho+b_0) (5)

C̃_t=tanh(X_t W_xc+H_(t-1) W_hc+b_c) (6)

P_t=P_t⊙P_(t-1)+I_t⊙C̃_t (7)

H_t=O_t⊙tanh(P_t) (8)

where I_t∈R^(n×h) denotes input gate, F_t∈R^(n×h) signifies forget gate, and O_t∈R^(n×h) denotes the output gate. X_t input info. W_xi, W_xf, W_xo, W_xc, W_hi, W_hf, W_ho, and W_hc are the weight parameters, and b_i, b_f, b_o, and b_c are bias limits. C̃_t and P_t symbolise current concealed state, H_t the candidate memory cells, and M_t the present memory cells. Following this, the LSTM cell calculates the states. Given the X_t and H_(t-1), It is possible to compute pertinent data, including input and forget states.

The state of the cell is generated by the input and forget gates. Subsequently, the subsequent LSTM cell is programmed with cell states and secret keys. Since there is among the previous state h_prev together with the current input x into the LSTM, and since there is no chance to communicate prior to the gate, contextual information may be lost as a result. We suggest a bidirectional Mogrifier LSTM that enhances both concept extraction and contextual modelling, drawing inspiration from the Mogrifier LSTM, which advances contextual modelling by supplying the interaction before the gate. BiMogrifier LSTM uses mutual gating to update both the input and the past hidden states. With every cycle, the crossed by the gate. The LSTM perfect extracts hidden information in both directions. h ⃗ and h in

the direction of travel forward and backward, individually. In order to process the input embedding and the past hidden states, the Mogrifier operation uses the variables xi and h ⃗_prev, respectively.

h ⃗=Mogrifier(x ⃗,h ⃗_prev) (9)

h ⃖=Mogrifier(x ⃖,h ⃖_prev) (10)

h_(BiMogrifier-LSTM)=[h ⃗,h ⃖] (11)

x^i=2σ(Q^i h_prev^(i-1))⊙x^(i-2),for odd i∈[1,…,r] (12)

h_prev^i=2σ(R^i x^(i-1)),for eveni∈[1…r] (13)

where Q^i and R^i are mediums with arbitrarily initialized standards, besides r signifies the sum of rounds. σ characterizes a logistic purpose and ⊙ characterizes an product.

3.5.1. Bidirectional LSTM

The second layer processes the attack detection using basic BiLSTM network. To train its models, BiLSTM networks take both local and global characteristics. The concealed representations, h ⃖ besides h ⃗ Equations (14) and (15), correspondingly. The hidden representation BiLSTM is gotten by h ⃖ besides h ⃗ as assumed in Equation (16).

h ⃗=LSTM[(h_1) ⃗,…,(h_n) ⃗] (14)

h ⃖=LSTM[(h_1) ⃖,…,(h_n) ⃖] (15)

h_BiLSTM=[h_1,…,h_n] (16)

The concatenation layer receives as input the contextual illustrations derived from the LSTM and the conventional BiLSTM.

3.5.2. Concatenation Layer

The representations from the levels below are added together element by element in this layer. The merging of two separate networks is symbolised by h_Final as shown in Equation (17).

h_Final=[h_(BiMogrifier-LSTM)+h_BiLSTM] (17)

3.5.3. Multihead Attention

In clinical reports, the entities do not exist in a vacuum; rather, they are interdependent and exhibit lengthy intervals between entity characteristics. In light of the gravity of this dependence, it is imperative that we pay closer attention to dependent characters by giving them larger weights to noteworthy characters and smaller weights to less significant ones. The model must enhance the significance of content-dependent characters in order to reflect this reliance.

In order to find relevant features, we employed an MHA technique. To calculate attention scores, we use Equation (18).

attention(Q,K,V)=SoftMax(〖QK〗^t/√d)V (18)

In order to focus on different parts of the value vector channels, MHA uses parallel h heads. In order to calculate self-attention, we set the Q, K, and V parameters, which stand for the sentence's characters, to equal. The parameters for learning are described as W_i^Q∈R^(n×d/h), W_i^K∈R^(n×d/h), and W_i^V∈R^(n×d/h), . The ith head attention is intended using Equation (19).

M_i=Attention(〖QW〗_i^Q, 〖KW〗_i^K, 〖VW〗_i^V) (19)

To get the tth character of the phrase's output, we use Equation (20), which involves concatenating the calculation output from these limits h times and then performing a linear transformation. where concat() denotes the splicing function besides W_i^Q∈R^(n×d/h)is the weight parameter.

M_t=concat(h_1,h_2,h_3,…,h_h) W^0 (20)

3.5.4. SoftMax

After MHA layer, we added SoftMax layer to decipher the anticipated labels. Equation (21) is used to get the evaluation score.

S(X,y)=∑_(i=1)^n▒ 〖P_i,y_i 〗 +∑_(i=1)^n▒W_(y_i,y_(i+1)) (21)

where X characterizes sequence, y characterizes the consistent label arrangement, P_(i,j) represents the character labelled as label j, W characterizes the transition matrix, and W_(y_i,y_(i+1)) is state from label i to j.

SoftMax to calculate conditional probability of the arrangement label y given X, using Equation (22).

P(y ⎹ X)=e^(S(x,y))/(∑_(y ̃∈Y_x)▒e^(S(x,y))) (22)

3.5.5. Fine-tuning using Election Optimizer Algorithm

Adjusting the parameters of the suggested classifier is done using the EOA in this study [29]. During the party nomination phase, the algorithm looks for varied solutions by exploring solution spaces; during the presidential election phase, it seeks the local optimum simultaneously. Starting with a set of randomly generated candidate solutions, the optimisation process begins. By iteratively applying a certain fitness function, these solutions are evaluated according to inherent optimisation guidelines for updates.

A) Initialization Model

When using EOA, the first step in candidate initialization is to generate a random collection of candidates that fall inside the lower limit (LB) besides upper bound (UB) of the strategy space, which is the solution space for the problem at hand.

X=[X_1,..,X_i,…,X_N]^T=[■(x_1,1&x_1,2&■(⋯&x_(1,j)&■(⋯&x_(1,Dim)))@:&:&■(:&:&■(:&:))@■(x_(i,1)@:@x_(N,1)

)&■(x_(i,2)@⋮@x_(N,2)
)&■(■(⋯@⋮@⋯)&■(x_(i,j)@⋮@x_(N,j)
)&■(■(⋯@⋮@⋯)&■(x_(i,Dim)@⋮@x_(N.Dim))))] (23)

where X_i mentions to the i-th applicant in set. N is total sum of candidates. Dim is sum of candidate keys, and x_(i,j) denotes j-th policy of applicant and is intended as:

x_(i,j)= 〖LB〗 _j+(〖UB〗 _j- 〖LB〗 _j)×rand (24)

where rand signifies a random sum ranging from 0 to 1. 〖LB〗 _j and 〖UB〗 _j signify lower besides upper bounds for the j-th plan within the policy space, correspondingly.

B) Party Nomination

In this portion, we'll look at EOA's party nomination behaviour, which is related to the optimisation process's exploration phase. To run for president, a candidate must first become the party's nomination. Two essential actions—referring to a plan (SR) and soliciting creative ideas from employees (ISST)—can accomplish this. Candidates' ability to compete for party nominations is intended to be boosted by these actions.

B.1.) Strategy Reference

Candidates engage in this practice when they modify their campaign tactics by making allusions to other party campaigners. The following equation for updating positions is suggested to make the SR behaviour easier.:

X_1 (t+1)=X_R (t)+(X(t)-X_best (t))×AF (25)

where X_1 (t+1) characterizes key for next repetition at the period t + 1 and designates the strategies. X_R (t) is an unknown candidate within the similar party who can provide insight into potential strategy shifts for the campaign, and X_best is indicative of the candidate who received the most votes in the i-th cycle. The following equation determines AF, a stochastic strategy reference factor:

AF=a×r_3.exp(1-t/T) (26)

where α is a subtle among 0 and 20. r_3 is a chance sum among 0 besides 1. T characterizes the extreme repetition sum, while t signifies the present restatement.

B.2.) Innovative Proposals of the Staff Team

This conduct necessitates that every candidate has their own staff group that comes up with fresh policy ideas. Candidates can increase their popularity in the polls by responding quickly to staff recommendations for policy changes that might help them run for office. This behaviour can be expressed mathematically in the following way:

X_2 (t+1)=X(t)×r_4+r_5×X_S (t) (27)

where r_4 and r_5 characterize random statistics among 0 besides 1. X_S (t) Symbolizes the innovative proposals put team, which are intended by using the subsequent equation:

X_S (t)=(UB-LB)×r_6+LB (28)

where r_6 This conduct necessitates that every candidate has their own staff group that comes up with fresh policy ideas. Candidates can increase their popularity in the polls by responding quickly to staff recommendations for policy changes that might help them run for office. This behaviour can be expressed mathematically in the following way:

C). Presidential Election

At the end of the exploration phase, the top party candidates battle even more fiercely, but the control factor SF ensures that the population remains diverse. Here we present the EOA exploitation phase, which is in line with the actions seen in the recent presidential election. During this stage of the campaign, candidates participate in a variety of activities, such as television debates (TD) with the goal of increasing their support in the polls. Incorporating these two behavioural traits—described in detail below—into the EOA model represents the process of the presidential election phase.

C.1.) Televised Debate

Candidates engage in this practice when they engage in debates with other candidates from the same party on television in an effort to boost their popularity. While preparing for the debate, candidates research the platforms of their opponents in order to adjust their own campaign plans. Below, you can find its mathematical expression.:

X_2 (t+1)=X_best (t)+r_7×(X_best (t)-TF×X_R (t)) (29)

where r_7 is a random charge among 0 and 1. X_bestt characterizes optimal voting strategies, while X_R represents the adversary's tactics from the opposing side. A value between 0 and 2 is randomly assigned to TF, and it shows how much the opponent's campaign policies are referenced.

C.2.) Campaign Speech

Candidates use campaign speeches to communicate their platform pledges to voters and adjust their tactics in response to people' political goals. A candidate's demeanour while making a campaign speech can be described in the following way:

X_4 (t+1)=X_best (t)+(X_M (t)-X_best (t)).SF×r_8 (30)

where X_M (t) is the average of all the solutions found up to this point in time for the t-th iteration, as determined by Equation (31). The value of r_8 might be anything from zero to one. The equation is used to determine the control factor SF= 〖(t+1)〗 ^(-(2.r_9+1)/ 〖(T+1)〗 ^2), where T is the maximum

repetition sum besides r_9 is a chance value among 0 besides 1.

X_M (t)=1/N ∑_(i=1)^N▒〖X_i (t)〗 (31)

4. RESULTS AND DISCUSSION

Here, we present the research, review the experimental setting, and compare the results of our attack detection and disease prediction classifiers using existing methodologies.

4.1. Experimental Environment

Many programmes and tools were used to build and evaluate the suggested model's performance. These included NumPy, Python, Skfeature, Scikit Learn, and Scikit. Furthermore, a device with an IntelI Core I i7-4790 CPU @ 3.60 GHZ besides 16 GB RAM was used for data sample organisation, algorithm application, and result interpretation [30].

4.2. Evaluation Metrics

We chose a set of presentation criteria that are well-known in the IDS to evaluate our model's effectiveness. Since Accuracy (ACC) is a simple metric for how well the model can identify instances, it was decided to use it as the main performance indication for the model. On the other hand, we incorporated detection rate (DR), besides F1 score (F1) to afford a more comprehensive picture of the representation's predictive capabilities. These criteria were providing a fair valuation of model's performance, taking into consideration the expenses associated with misclassification. Since false alarms can cause financial and operational disruptions in the IoMT setting, the FPR takes on added significance in this context. For accurate intrusion detection, the DR (or recall) is crucial, and the F1 score strikes a good compromise between the model's specificity and sensitivity by providing a harmonic mean of recall and precision.

ACC=(TP+TN)/(TP+TN+FP+FN) (32)

FPR=FP/(TN+FP) (33)

DR=TP/(TP+FN) (34)

F1=TP/(TP+0.5*(FP+FN)) (35)

where TP,TN,FP,and FN signify true positive, true negative, false positive, besides false negative, individually.

4.3. Validation Analysis of Proposed Attack Classifier on Diverse Ratio of Data

Table 2 and Figure 2 provides the experimental analysis of proposed attack classifier on combined datasets in terms of various metrics on different ratio of data.

Table 2: Validation Analysis of Proposed Classifier on both Datasets.

Phases	Accuracy (%)	Recall (%)	Precision (%)	F-Score (%)
60-40	83.18	83.03	89.21	86.0
70-30	96.10	97.06	95.45	96.25
80-20	99.03	99.81	98.7	99.25

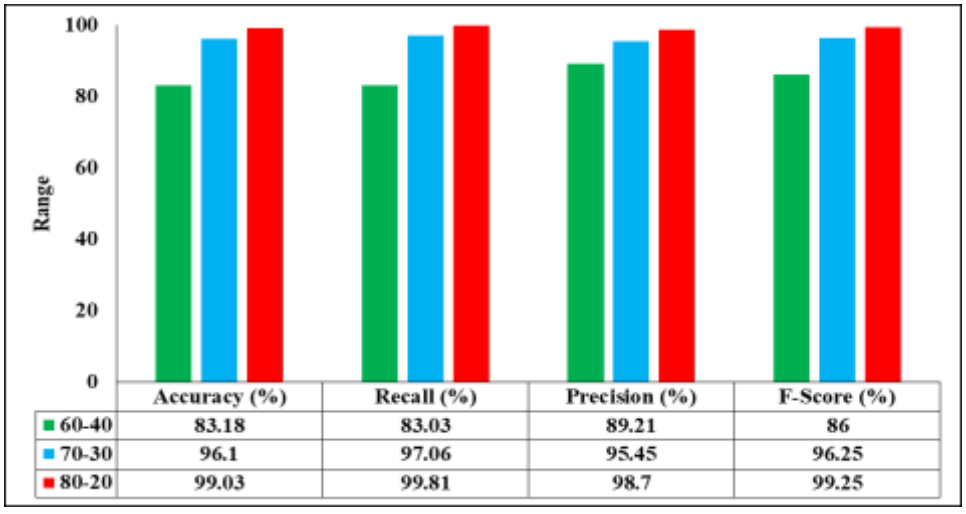

Figure 2: Visual Representation of Different Ratio of Data on Projected Model.

In the Representation of diverse ratio of data on projected model as the 60-40 data split-up phase, the accuracy as 83.18 and recall as 83.03 also precision as 89.21 also f1-score as

86.0 correspondingly. Then the 70-30 data split-up phase, the accuracy as 96.10 and recall as 97.06 also precision as 95.45 also f1-score as 96.25 correspondingly. Then the 80-20 data split-up phase, the accuracy as 99.03 and recall as 99.81 also precision as 98.7 also f1-score as 99.25 correspondingly.

4.4. Validation Analysis of Proposed Disease Detection Classifier

Figure 3 provides the experimental analysis of hybrid pre-trained classifier on combined datasets in terms of various metrics on different ratio of data.

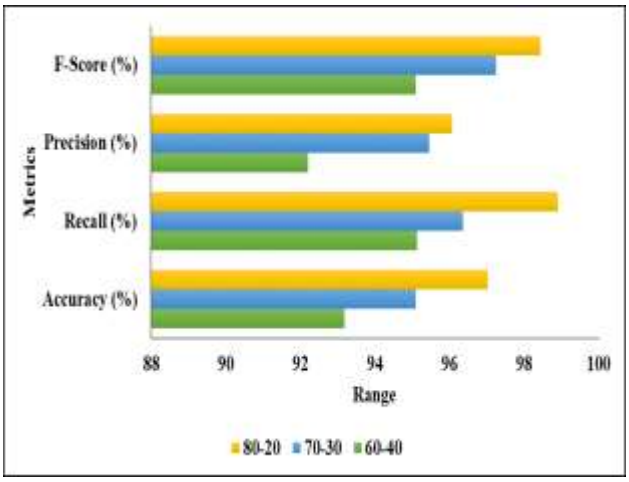

Figure 3: Training besides Testing Data Analysis on Hybrid Pre-trained Models.

Training and Testing Data Analysis on Hybrid Pre-trained models as in the 60-40 data phase as the accuracy of 93.18 also recall as 95.13 also precision of 92.21 also f1-score as 95.10 correspondingly. After the 70-30 data phase as the accuracy of 95.10 also recall as 96.36 also precision of 95.45 also f1-score as 97.25 correspondingly. After the 80-20 data phase as the accuracy of 97.03 also recall as 98.91 also precision of 96.07 also f1-score as 98.45 correspondingly.

A) Validation analysis of Proposed Optimization (CSO-MA)

The presentation of proposed optimizer is tested with different learning rate such as 0.1, 0.01, 0.001 and 0.0001 on both datasets, which is exposed in Figure 4.

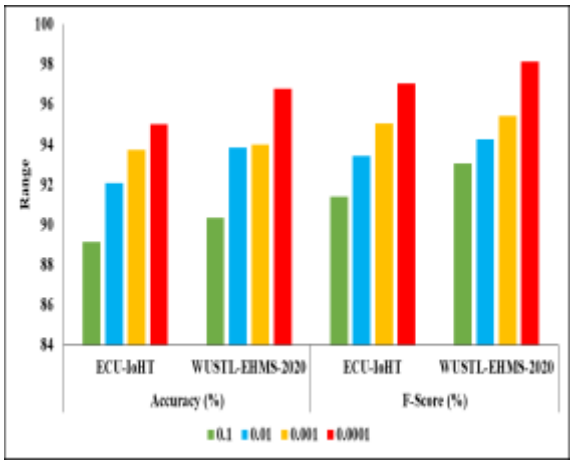

Figure 4: Visual Representation of Learning Rate Analysis of CSO-MA.

In the Representation of Learning Rate Analysis of CSO-Maas in the 0.1 learning rate of ECU-IoHT accuracy as 89.15 also WUSTL-EHMS-2020 technique accuracy of 90.35 also the ECU-IoHT technique f-score as 91.41 also WUSTL-EHMS-2020 technique f-score as 93.07 correspondingly.

Then the 0.01 learning rate of ECU-IoHT accuracy as 92.08 also WUSTL-EHMS-2020 technique accuracy of 93.86 also the ECU-IoHT technique f-score as 93.45 also WUSTL-EHMS-2020 technique f-score as 94.25 correspondingly. Then 0.001 learning rate of ECU-IoHT accuracy as 93.73 also WUSTL-EHMS-2020 technique accuracy of 94.01 also the ECU-IoHT technique f-score as 95.07 also WUSTL-EHMS-2020 technique f-score as 95.45 correspondingly.

Then the 0.0001 learning rate of ECU-IoHT accuracy as 95.03 also WUSTL-EHMS-2020 technique accuracy of 96.79 also the ECU-IoHT technique f-score as 97.05 also WUSTL-EHMS-2020 technique f-score as 98.14 correspondingly.

4.5. Comparative Analysis on Proposed Attack Classifier on First dataset

The presentation of proposed BDMLSTM is tested with existing techniques such as LinSVM [27], CNN [26], LSTM [23], CGAN-CNN [24], CNN-LSTM [29], FusionNet [32] and PLSTM [34] in terms of different metrics on both datasets. The existing representations uses different datasets; therefore, the research work implements basic models of existing papers and results are averaged that is shown in Figure 5 to 7.

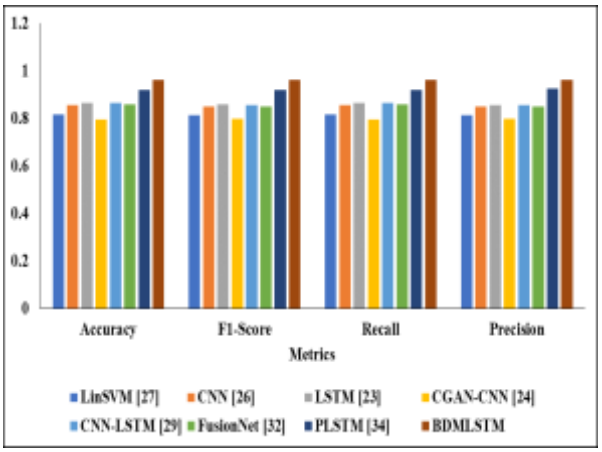

Figure 5: Comparative Analysis of Diverse Representations on First Dataset.

And also the Comparative analysis of diverse models on First dataset. In the analysis of LinSVM [27] technique accuracy as 0.8153 also f1-score as 0.8134 also recall as 0.8153 also the precision of 0.8117 correspondingly. Then CNN [26] technique accuracy as 0.8549 also f1-score as 0.8504 also recall as 0.8549 also precision as 0.8479 correspondingly. Then LSTM [23] technique accuracy as 0.8628 also f1-score as 0.8573 also recall as 0.8628 also the precision of 0.8551 correspondingly. Then CGAN-CNN [24] technique accuracy as 0.7942 also f1-score as 0.7962 also recall as 0.7942 also the precision of 0.7984 correspondingly. Then CNN-LSTM [29] technique accuracy as 0.8628 also f1-score as 0.8554 also recall as 0.8628 0.8538 correspondingly. Then FusionNet [32] technique accuracy as 0.8575 also f1-score as 0.8499 also recall as 0.8575 correspondingly. Then PLSTM [34] technique accuracy as 0.9179 also f1-score as 0.9185 also recall as 0.9179 also the precision of 0.9245 correspondingly. BDMLSTM technique accuracy as 0.9607 also recall as 0.9607 also recall as 0.9607 also the precision of 0.9614 correspondingly.

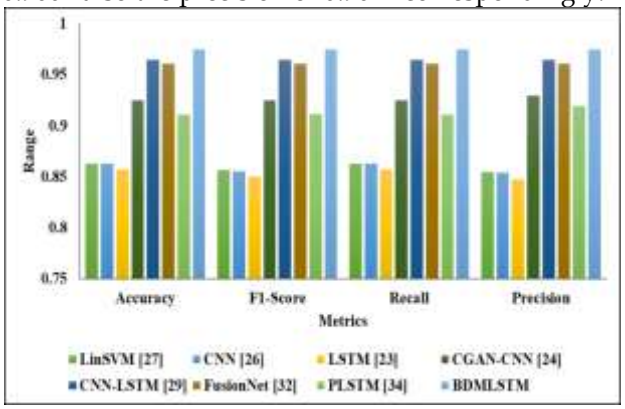

Figure 6: Comparative Analysis of Innumerable Classifiers on Second Datasets.

In the Comparative Analysis of numerous classifiers on second datasets as the LinSVM [27] technique accuracy as 0.8628 also f1-score as 0.8564 also recall as 0.8628 also the precision of 0.8544 correspondingly. Then CNN [26] technique accuracy as 0.8628 also f1-score as 0.8554 also recall as 0.8628 also the precision of 0.8538 correspondingly. Then LSTM [23] technique accuracy as 0.8575 also f1-score as 0.8499 also recall as 0.8575 also the precision of 0.8478 correspondingly. Then CGAN-CNN [24] technique accuracy as 0.925 also f1-score as 0.9251 also recall as 0.925 also the precision of 0.9293 correspondingly. Then CNN-LSTM [29] technique accuracy as 0.9643 also f1-score as 0.9643 also recall as 0.9643 also the precision of 0.9643 correspondingly. Then FusionNet [32] technique accuracy as 0.9607 also f1-score as 0.9607 also recall as 0.9607 also the precision of 0.961 correspondingly. Then PLSTM [34] technique accuracy as 0.9107 also f1-score as 0.9115 also recall as 0.9107 also the precision of 0.9192 congruently. Then BDMLSTM technique accuracy as 0.975 also f1-score as 0.975 also recall as 0.975 also the precision of 0.975 congruently.

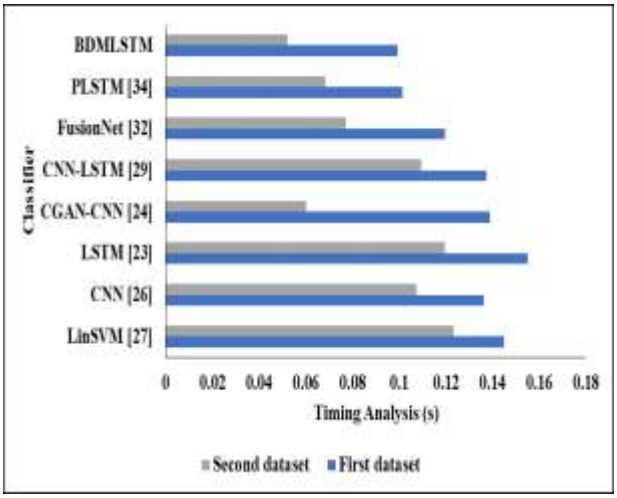

Figure 7: Timing Analysis of Proposed Model on Both Datasets.

In the time analysis of proposed classical with two dataset presentation as LinSVM [27] technique time analysis of first dataset rate as 0.1449 also second dataset rate as 0.1232 correspondingly. Then the CNN [26] technique time analysis of first dataset rate as 0.1363 also second dataset rate as 0.1073 correspondingly. Then the LSTM [23] technique time analysis of first dataset rate as 0.155 also second dataset rate as 0.1196 correspondingly. Then the CGAN-CNN [24] technique time analysis of first dataset rate as 0.139 also second dataset rate as 0.0602 correspondingly. Then the CNN-LSTM [29] technique time analysis of first dataset rate as 0.1373

also second dataset rate as 0.1095 correspondingly. Then the FusionNet [32] technique time analysis of first dataset rate as 0.1196 also second dataset rate as 0.0773 correspondingly. Then the PLSTM [34] technique time analysis of first dataset rate as 0.1013 0.0682 correspondingly. Then the BDMLSTM technique time analysis of first dataset rate as 0.0993 also second dataset rate as 0.0523 correspondingly.

5. DISCUSSION

Depending on the attack label, our model's performance varied in our investigation. The traits shared by various forms of assaults could explain this. As an example, the model may learn and identify certain patterns in assaults more easily, leading to better performance metrics for specific attack labels. However, not all assaults display obvious patterns; in fact, some may mimic legitimate traffic in order to evade detection and reduce performance metrics.

Computer power and time are essential for training convolutional neural networks (CNNs), particularly deep ones. Because of the need for precise hyperparameter adjustment and, at times, lengthy training durations, LSTMs can be difficult to train. Improving the security of IoMT schemes is possible through the optimisation of deep learning models to identify vulnerabilities posed by anomalies in real-time. Predictive healthcare services can save healthcare costs by reducing the need for costly emergency treatments besides hospitalisations by anticipating and preventing serious health issues.

The distribution of various assault labels in our dataset could possibly be influencing the variations. There may not be enough instances for the model to learn from if the dataset does not adequately reflect specific attack labels. This could lead to inaccurate attack identification. These differences in model performance emphasise the value of a diversified training sample. The training data for an intrusion detection system could not be representative of the real-world threats that the system faces. Because of this, it is crucial to train and update models continuously as new attack patterns appear.

Our study's dataset may not be representative of the variety of IoMT devices and network patterns in different contexts because it was derived from a single IoMT scenario. Because of this, our model may be biassed; for example, it may perform well on comparable data but poorly when forced to simplify for use in other IoMT settings. Our algorithm may have unknowingly picked up on the dataset's inherent biases or mistakes, which would have compromised its accuracy and dependability.

Datasets gathered from a variety of IoMT settings could be useful for future research. In doing so, the anomaly detection system would have a better, more complete picture of the traffic flowing via IoMT networks. The use of several datasets also has the ability to reveal and correct for any biases or inaccuracies that may exist in any one dataset. Improving model performance is the main reason why longer training epochs are important for DL architectures. The model can learn more from the data and minimise the loss function by adjusting its weights and biases over a longer period of time (epochs). Because training for an excessive number of epochs might lead to the training data and poor performance on unknown data, it is crucial to keep an eye out for overfitting symptoms in the model.

A broader variety of IoMT-related elements in training set could potentially be considered in future investigations as well. For instance, researchers could incorporate variables such as network traffic, device kind, manufacturer, version; variables pertaining to user behaviour, such as the timing and frequency of device usage; and variables pertaining to the environment, such as network conditions. With these extra capabilities, the intrusion detection scheme may be better presented and a more complete picture of the IoMT environment may be provided.

6. CONCLUSION

In this work, we introduce an innovative technique for the IoMT-specific extraction and categorization of clinical concepts for use in attack and disease prediction. Our experimental results show that the model outperforms existing IDS representations in important metrics like rate, besides F1 score, demonstrating the effectiveness of its dynamic adjustment of training epochs stopping. Notably, these intrusion detection developments improve the security framework necessary for dependable healthcare operations, which directly contributes to the larger IoMT context. Healthcare practitioners may rest certain that vital medical devices and patient data will remain secure and accessible because to our system's capacity to detect intrusions swiftly and correctly. At first, a hybrid pre-trained model is used for disease prediction, and then CSO-MA is used for fine-tuning. Bi-Mogrifier LSTM besides BiLSTM are the two systems that identify the assaults. The recommended model outdoes the state-of-the-art representations on both benchmark datasets. Improving our model's ability to process real-time data is essential for reducing reaction times to possible assaults and ensuring the uninterrupted provision of healthcare services. Finally, as the

Internet of Medical Things (IoMT) keeps expanding, researchers will have to figure out how to keep sensitive health information private while also warding off new dangers. Modern healthcare delivery and patient safety depend on the IoMT infrastructure, which can only be adequately protected by ongoing research and development of strong and flexible security solutions.

Acknowledgments:

Author Contributions: Veena R S: Conceptualization, Methodology, Writing – Original Draft - Nandini Prasad K S: Supervision, Project Administration, Resources - Mesith Chaimanee: Formal Analysis, Validation - Mithileysh Sathiyanarayanan: Software, Visualization - Asokan Vasudevan: Review & Editing, Funding Acquisition- Divyashree H B: Data Curation, Investigation - Nasyra Ab. Jamil: Writing – Review & Editing, Methodology, All authors have read and approved the final manuscript and agree to be accountable for all aspects of the work.

REFERENCES

Ahmed, M.; Byreddy, S.; Nutakki, A.; Sikos, L.; Haskell-Dowland, P. ECU-IoHT, 2020. Available online: https://doi.org/10.25958/5f1f97b837aca.

Ahmed, M.; Byreddy, S.; Nutakki, A.; Sikos, L.F.; Haskell-Dowland, P. ECU-IoHT: A dataset for analyzing cyberattacks in internet of health things. Ad Hoc Netw. 2021, 122, 102621.

Alalhareth, M. & Hong, S.-C. (2024). Enhancing the Internet of Medical Things (IoMT) Security with Meta Learning: A Performance Driven Approach for Ensemble Intrusion Detection Systems. Sensors, 24(11):3519.

Alalhareth, M., & Hong, S. C. (2023). An Adaptive Intrusion Detection System in the Internet of Medical Things Using Fuzzy-Based Learning. Sensors, 23(22), 9247.

Al-Hawawreh, M., & Hossain, M. S. (2023). A privacy-aware framework for detecting cyber attacks on internet of medical things systems using data fusion and quantum deep learning. Information Fusion, 99, 101889.

Alzahrani, A. A. (2023). Using Artificial Intelligence and Cybersecurity in Medical and Healthcare Applications. Alzahrani, AA.

Alzubi, J. A., Alzubi, O. A., Qiqieh, I., & Singh, A. (2024). A blended deep learning intrusion detection framework for consumable edge-centric iomt industry. IEEE Transactions on Consumer Electronics, 70(1), 2049-2057.

Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A. V., & Di Franco, F. (2023). The role of machine learning in cybersecurity. Digital Threats: Research and Practice, 4(1), 1-38.

Arif, F., Khan, N. A., ul Haq, Q. M., Asim, M., & Ahmad, S. (2024). An sdn-ai-based approach for detecting anomalies in imbalance data within a network of smart medical devices. IEEE Consumer Electronics Magazine, 13(6), 28-36.

Chakraborty, C., Nagarajan, S. M., Devarajan, G. G., Ramana, T. V., & Mohanty, R. (2023). Intelligent ai-based healthcare cyber security system using multi-source transfer learning method. ACM Transactions on Sensor Networks.

Cheng, R. & Jin, Y. A competitive swarm optimizer for large scale optimization. IEEE Trans. Cybern. 45, 191–204 (2015).

Devi, V. A., Thirumalraj, A., Kavin, B. P., & Seng, G. H. Securing the Predicted Disease Data using Transfer Learning in Cloud-Based Healthcare 5.0. In Intelligent Systems and Industrial Internet of Things for Sustainable Development (pp. 101-117). Chapman and Hall/CRC.

Dhanushkodi, K., Shunmugiah, J., Velladurai, S. M., & Rajendran, S. (2024). An optimal attention PLSTM□ based classification model to enhance the performance of IoMT attack detection in healthcare application. Transactions on Emerging Telecommunications Technologies, 35(6), e5008.

Faisal, R. H., Debnath, S., Islam, M. M. U., Sifath, S., Kakon, S. A., Alam, M. S., & Siddique, N. (2023). A modular fuzzy expert system for chemotherapy drug dose scheduling. Healthcare Analytics, 3, 100139.

Franklin, E., & Pranggono, B. (2023). Anomaly-Based Intrusion Detection System for the Internet of Medical Things. IJID (International Journal on Informatics for Development), 12(2), 374-385.

Kanagala, P. (2023). Effective cyber security system to secure optical data based on deep learning approach for healthcare application. Optik, 272, 170315.

Khatun, M. A., Memon, S. F., Eising, C., & Dhirani, L. L. (2023). Machine Learning for Healthcare-IoT Security: A Review and Risk Mitigation. IEEE Access.

Kilincer, I. F., Ertam, F., Sengur, A., Tan, R. S., & Acharya, U. R. (2023). Automated detection of cybersecurity attacks in healthcare systems with recursive feature elimination and multilayer perceptron optimization. Biocybernetics and Biomedical Engineering, 43(1), 30-41.

Kumar, A. K., Vadivukkarasi, K., & Dayana, R. (2023, February). A novel hybrid deep learning model for botnet attacks detection in a secure iomt environment. In 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS) (pp. 44-49). IEEE.

Mijwil, M., Salem, I. E., & Ismaeel, M. M. (2023). The significance of machine learning and deep learning techniques in cybersecurity: A comprehensive review. Iraqi Journal For Computer Science and Mathematics, 4(1), 87-101.

Praveena Anjelin, D., & Ganesh Kumar, S. (2024). An effective classification using enhanced elephant herding optimization with convolution neural network for intrusion detection in IoMT architecture. Cluster Computing, 27(9), 12341-12359.

Shambharkar, P. G., & Sharma, N. (2024). Deep learning-empowered intrusion detection framework for the Internet of Medical Things environment. Knowledge and Information Systems, 66(10), 6001-6050.

Srinivasan, V., Raj, V. H., Thirumalraj, A., & Nagarajan, K. (2024). Original Research Article Detection of Data imbalance in MANET network based on ADSY-AEAMBi-LSTM with DBO Feature selection. Journal of Autonomous Intelligence, 7(4), 1094.

Sun, Z., An, G., Yang, Y., & Liu, Y. (2024). Optimized machine learning enabled intrusion detection 2 system for internet of medical things. Franklin Open, 6, 100056.

Thirumalraj, A., Asha, V., & Kavin, B. P. (2023). An Improved Hunter-Prey Optimizer-Based DenseNet Model for Classification of Hyper-Spectral Images. In AI and IoT-Based Technologies for Precision Medicine (pp. 76-96). IGI g.

Zhang, X., Zheng, X., Cheng, R., Qiu, J., & Jin, Y. (2018). A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. Information Sciences, 427, 63-76.

Zhang, Y. (2024). Intrusion detection using a SV-SMOTE method in IoMT. International Journal of Modelling, Identification and Control, 44(4), 294-302.

Zhou, S., Shi, Y., Wang, D., Xu, X., Xu, M., & Deng, Y. (2024). Election Optimizer Algorithm: A New Meta-Heuristic Optimization Algorithm for Solving Industrial Engineering Design Problems. Mathematics, 12(10), 1513.

Zukaib, U., Cui, X., Zheng, C., Hassan, M., & Shen, Z. (2024). Meta-IDS: Meta-learning-based smart intrusion detection system for Internet of Medical Things (IoMT) network. IEEE Internet of Things Journal, 11(13), 23080-23095.

Zukaib, U., Cui, X., Zheng, C., Liang, D., & Din, S. U. (2024). Meta-Fed IDS: Meta-learning and Federated learning based fog-cloud approach to detect known and zero-day cyber attacks in IoMT networks. Journal of Parallel and Distributed Computing, 192, 104934.