Journal of Theoretical and Applied Information Technology
30st June 2026. Vol.104. No.12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

HIERARCHICAL EXPLAINABLE DEEP NEURAL ARCHITECTURE FOR REAL-TIME INTRUSION DETECTION IN ENGLISH EDUCATION ECOSYSTEMS

DR S. REMA DEVI[1], DR. KHURSHEEDA KHATOON[2], VAMSIDHAR TALASILA[3], VIMOCHANA.M[4], DR. PUNIT PATHAK[5], DR. R. BALAKRISHNA[6], DR.M.SHYAMALA BHARATHY[7]

[1]Associate Professor of English, Head - Department of S & H, SRM Madurai College for Engineering and Technology, Pottapalayam, Sivagangai District – 630612, India
[2]Language Instructor, English Department, Jazan University, KSA.
[3]Assistant Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, A.P., India.
[4]Assistant Professor, Department of English, Panimalar Engineering College, Poonamallee, Chennai, India.
[5]Assistant Professor , School of Liberal Arts and Human Sciences, Auro University, Gujarat, India.
[6]Associate professor, Department of Computer Science and Engineering, Vels Institute of Science Technology and Advanced Studies(VISTAS), Chennai, India.
[7]Associate Professor, AMET University, Department of Nautical Science, Kanathur, ECR, Chennai – 604112, India
[1]remagopu@gmail.com, [2]kkhatoon@jazanu.edu.sa, [3]talasila.vamsi@kluniversity.in, [4]vimvijay210@gmail.com, [5]pathakpunit102@gmail.com, [6]r.balakrishna1989@gmail.com, [7]shyamalabharathym@ametuniv.ac.in.

ABSTRACT

Some of the online English education systems that have become highly vulnerable to advanced cyber attacks threatening to compromise personal academic and administrative information include learning management systems (LMS), virtual classrooms, student portals, and institutional networks just to mention a few. Due to class imbalance, poor feature representation, and lack of interpretability, the traditional intrusion detection systems (IDS) that use shallow machine learning or the standard deep learning models often cannot detect the unusual types of attacks, which reduces the trust in automated security decisions. The proposed solution to these issues in this work is a Hierarchical Explainable Deep Neural Architecture (HEDNA) of real-time intrusion detection in an English teaching environment. The proposed system integrates explainable AI methods, explainable AI methods, oversampling and cost-sensitive learning to reduce class imbalance and multi-level hierarchical feature engineering to generate intelligible security decisions. Network traffic characteristics such as flow duration, number of packets, rate of the bytes and indicators of session-level behavior are used to identify low-level and contextual trends by hierarchically aggregating the characteristics. The model was tested using the TII-SSRC-23 dataset that contains realistic benign and malicious network flows and was implemented in Python. It is a suitable architecture to be deployed in real-time since the experimental results indicate that the proposed architecture can achieve an accuracy of 96% in terms of intrusion detection with low false-positive rates and latency. Also, explainability modules provide useful information to the administrators by highlighting significant contributing traffic characteristics. The results confirm that explainable AI and hierarchical deep learning have a significant impact on enhancing accuracy, transparency, and operation feasibility in the protection of modern English education ecosystems.

Keywords: *Hierarchical Deep Neural Network, Explainable AI, Real-Time Intrusion Detection, English Education Ecosystems, TII-SSRC-23 Dataset*

1. INTRODUCTION

The rapid development of online learning makes the process of English learning a highly digital setting that relies on learning management systems, cloud-based learning materials, virtual learning classrooms, online examinations, student databases, and real-time communication platforms[1]. Although these technologies increase the effectiveness and accessibility of education, they also expose educational institutions to serious cybersecurity risks. Cybercriminals have found school networks to be

an appealing target since they contain sensitive information on students, their grades, logins, institutional messages, and managerial figures [3]. [2] Data leaks, service interruptions, academic fraud, and a decline of institutional trust would result from a successful cyberattack on such systems. Currently, the development of intelligent and reliable intrusion detection techniques is crucial to ensuring safe and ongoing digital learning environments[4].

The intrusion detection systems that are commonly used are conventional methods based on traditional machine learning models such as the Random Forest, Support Vector Machines, and shallow neural networks[5]. Because of their inability to model complex relationships between traffic and dynamic attack patterns, such techniques often fail in the real-world, although they have shown good performance on benchmark datasets[6]. Also, the distinction between benign and malicious activity is harder because educational network traffic is highly dynamic and it contains multimedia streaming, assignment submissions, numerous logging-in sessions, forum posts, and updates of the background system[7]. [9]Moreover, most publicly available intrusion datasets are highly unbalanced with infrequent attack types underrepresented and benign traffic predominant[8]. Due to this imbalance, IDS models make bias predictions, which lead to large false negative and failure to detect sophisticated attacks such as botnet activity, low rate brute force or hidden reconnaissance attacks[10].

Deep learning networks such as CNNs, RNNs, BiLSTM, and transformer based networks have improved the performance of intrusion detection by learning hidden representations of the network flow in high dimensions. However, most of the deep models are black-box systems that do not explain their decisions. In education in particular, where administrators require understandable notifications to respond to mitigate measures in real-time, this non-transparency reduces trust and limits its real-world application. Moreover, leaders of the network behaviors such as low-level pattern of packet, flow-level statistics, and session-level semantics should be analyzed concurrently is disregarded by most of the current deep intrusion detection systems, which treat all of the retrieved traffic information as equal. There is an urgent need hence an intrusion detection structure that (i) has a high detection rate of both common and uncommon attacks, (ii) is well resistant to imbalance in the classes, (iii) is real-time and has a low-latency performance, and (iv) is explainable to aid administrator decision-making.

To address these issues, this work proposes a solution of Hierarchical Explainable Deep Neural Architecture (HEDNA) of real-time intrusion detection in English education ecosystems. The proposed technique is a combination of packet-based indications, flow-based metrics, and session-based behavioral patterns to build a hierarchical feature engineering mechanism to represent multi-level representations of traffic. In order to increase the minority attacks classification, the method also incorporates class imbalance strategies such as cost sensitive learning and SMOTE-based oversampling. In order to enhance trust and transparency, explainable AI techniques such as SHAP and LIME are applied to identify the most critical factors that contribute to each intrusion decision. The architecture is evaluated using the current TII-SSRC-23 dataset that provides realistic benign and malicious traffic samples that are indicative of the current trends in cyberattacks.

1.1 Problem Statement

Therefore, the challenge is to develop an intrusion detection framework that will precisely detect common and rare attacks and give interpretable predictions in real time with efficacy within heterogeneous, sensitive, and dynamic English education ecosystems[11]. Addressing these challenges is the key to business continuity, preserving instructional resources, and developing trust in online education. Learning environments that are online have also increased greatly, and are considered an excellent target of cyberattacks, which attack personal student, faculty, and administrative data [12]. The existing intrusion detection system, especially the traditional machine learning or shallow neural networks-based intrusion detection systems, failed to detect sophisticated attacks due to poor representation of features, absence of temporal analysis, and improper management of the problem of class imbalance. The result was that rare types of attacks were not detected and high rate of false positive resulted into unnecessary network intervention. Also, most of the models were not interpretable and that is why the administrators were not able to accept the forecasts or take relevant action. Real-time detection was not possible because of the huge computing cost and latency.

1.2 Motivation of the Study

This has been inspired by the increasing use of online learning tools, which pose an attack point in networks and enable malicious attackers to exploit various vulnerabilities. The problems that most traditional intrusion detection algorithms cannot handle in real life are class imbalance, interpretability and latency limits. Since this has led to educational networks being complex systems that comprise a diverse data type, such as administrative transactions, multimedia material, and interaction sessions. Operational security needs efficient intrusion detection systems that can be interpreted. Although they disclose multi-level connections and generate predictable forecasts, hierarchical deep learning structures have proved to be effective in several areas. However, it was not shown that they were ever used in schools. This research paper is used to address this major gap in research by employing explainable modules, hierarchical feature engineering, and real-time processing. The proposed solution provides useful information to the system administrators in addition to identifying common and uncommon threats in a more effective manner. This is a practical and important motive in the modern digital learning ecosystems since it conforms to the need to ensure the security of learning environments without disrupting the smooth running of operations.

1.3 Significance of the Study

The importance of the research study is that it can be used to improve security of cyber in the English learning institutions that are becoming more susceptible to advanced attacks. The accuracy of detection and interpretability were ensured based on explainable AI techniques and deep neural networks, as well as hierarchical feature engineering. The framework better detects the unusual types of attacks that traditional models failed to detect as it overcame the problem of class imbalance. Real-time deployment capability ensured that threats were mitigated in time without disrupting the ongoing activities in the network or compromising instruction. Also, the research provided a systematic way of extracting features, model training and pre-processing that can be used to various digital platforms. The proposed methodology was strong and had a high level of practicality based on the TII-SSRC-23 dataset that represents real-life traffic trends in educational networks. The report also contributed to the literature and provided the foundation of future research by illuminating hierarchical modelling with explainable AI in cybersecurity.

1.4 Key Contributions

- For precise real-time intrusion detection in educational networks, a hierarchical explainable deep neural architecture was created.
- Multi-level feature engineering was performed by integrating different class imbalance management and explainable AI modules to obtain better interpretability and results.
- Used TII-SSRC-23 dataset, representing benign and harmful network flows of English education ecosystems in 2023.
- Compared to the baseline and state-of-the-art methods, the detection rate was high (96), few false positives, and successful detection of abnormal attacks.

The subsequent sections of the paper have been organized as follows: A summary of the intrusion detection literature and an exposition of the drawbacks associated with previous approaches are provided in Section 2. The details related to the proposed hierarchical and explainable deep neural approach, such as preprocessing, features, model design, and training procedure, are elaborated in Section 3. Experimental methodology, datasets used, evaluation metrics employed, results obtained, and visual illustrations are presented in Section 4. Section 5 discusses the findings, performance analysis, and comparisons using baseline methods. Section 6 concludes the study by highlighting the limitations and stating directions for future research in real-time intrusion detection within educational ecosystems.

2. RELATED WORKS

R.-K. Sheu [13] introduced will contribute to the field of Explainable AI in medicine by illuminating AI-based decisions concerning their ethical, legal, and clinical reliability. It utilizes an all-at-once survey approach that includes preprocessing-based explainability, knowledge distillation, interpretable machine learning, case studies and human-in-the-loop evaluation systems. Among the strengths identified in the work, it is possible to distinguish an increase in transparency, the development of a more reliable relationship with a clinician, systematic scoring, and an increase in human-machine interaction. Nevertheless, it also highlights other difficulties like subjective interpretation of explanations, little standardization of the scoring process, and the

impossibility of having accurate models along with interpretability. On the whole, the method provides valuable information but cannot be considered to be universal enough in different medical cases.

R. Dwivedi et al.,[14] purpose of this work is to provide a guideline to practitioners on how to select the appropriate explainable AI methodologies, surveying the key concepts of XAI, programming approaches, and development stages. It employs a review approach based on a taxonomy and classifies state-of-the-art XAI frameworks, toolkits and techniques and demonstrates them with examples. Its key benefits are the ability to have a better grasp of the XAI options, enhance the level of model transparency, as well as simplify the choice of the framework to be used in the real-life scenarios. Nevertheless, other shortcomings of the study include lack of universal XAI standards, variation in the performance of the tools across domains, and balancing interpretability with model complexity, which may limit the easy use in highly sensitive environments.

G. Vilone[15] present to provide an organization and clarification of the fast-growing field of Explainable AI through the systematic categorization of available theories, concepts, and assessment methods into a hierarchical structure. Through a systematic literature review, it classifies explainability techniques into human-centered and objective measures of metrics, pointing to what makes an explanation understandable and usable by its users. The strengths of this piece of work are in providing a single framework of knowing about XAI, determining the main specifications of effective explanations, and the direction of researchers to the corresponding substantive evaluation methods. Nevertheless, there are significant weaknesses, including the fact that there is no universal agreement about what was considered to be a valid explanation or there are no uniform evaluation criteria, which creates ambiguity in reliability and interpretability among systems.

R. Ghnemat,[16] proposed research will fill in the black-box shortcoming of deep learning by creating an explainable AI model on medical image classification that can disclose the process of decision making. The approach consists of division of the medical pictures in order to extract the parts that affect the predictions in order to achieve clearer understanding of the behavior of the model. Its merits are that it is more accurate, less time complex, and the additional transparency that enhances reliable clinical diagnosis. Yet, there are still certain limitations, including the use of rather small datasets, which may vary between various imaging settings, and the necessity to retain high interpretability rates without compromising the performance of the models. In spite of these limitations, the method enhances the reliability of the diagnosis as well as the trust of the users.

G. Marín Díaz, [17] proposed research will be focused on applying XAI, fuzzy C-means, and the Analytic Hierarchical Process (AHP) to improve transparent and data-driven decision making in B2B customer service. The approach divides customers according to patterns of interactions, clarifies model forecasts with the XAI instruments, and prioritizes the decision criteria with AHP to coordinate the support activities with the business objectives. Its benefits are the enhanced customer segmentation, insights that can be interpreted, and better resource allocation. Nevertheless, the method requires good quality data of interaction, might be computationally intensive, and might have scaling problems in application to rapidly evolving customer landscapes. These restrictions notwithstanding, it enhances strategic decision making and makes it transparent and accurate.

V. Bento, [18] improve the rate of image classification in the case of overlaid images, which include logos or text, and impair the performance of deep learning. Its approach consists of Explainable AI (in this case, Layer-Wise Relevance Propagation) to identify regions of the dataset that have misleading information to the classifier, and then preprocesses to eliminate such undesired overlays (by using cropping or generative inpainting) and retrains the model. Its benefits are the enhanced model understandability, great performance benefits, and its application in a range of imaging tasks. The method is, however, more computationally intensive and might not be able to handle complicated overlays that are hard to eliminate. In general, the workflow enhances the quality of classification, as well as, improves the reliability of models.

J. Shin, [19] intend to enhance the amplification of hand-gesture recognition with multichannel sEMG signal by resolving unstable prediction and poor time-varying features extraction of the signal. In this approach, a multi-stream deep learning that consists of Bi-TCN, CNN-SE blocks, and TCN-BiLSTM branches to learn long-term, spatial-temporal, and the pattern

of bi-dir gestures are introduced, and fused outputs are refined through channel attention. Its benefits are that it is highly accurate, its feature extraction is effective and its robustness is high in a variety of sEMG datasets and thus it can be used in prosthetics and human-machine interfaces. Nevertheless, the model needs to be calculated using multiple branches concurrently and can be prone to noise or low-quality sEMGs.

B. Pradhan,[20] proposed a develop more accessible and precise susceptibility maps of flooding by utilizing explainable AI to address the lack of transparency of deep learning. It applies a CNN model and SHAP to visualize the effect of the variables on flood prediction in Jinju Province, and it is highly accurate. The method improves the knowledge of the major variables such as land use and soil properties that can guide the stakeholders to make proper decisions. Its benefits are enhanced interpretability, high predictive accuracy and confidence in the machine-learned results. It can be rather computationally intensive, however, and needs specialized skills, which restricts its portability in resource-strained areas.

D. Bhati,[21] purpose of this work is to enhance the transparency and trust in the use of deep learning models in medical imaging by understanding interpretability and visualization methods related to these models and how they arrive at a decision. It discusses a variety of approaches to analyze and visualize the behavior of internal models and assist clinicians in knowing the importance of features and the logic of predictions. Its main benefit is that it will reduce the clinical usability gap between complex AI systems and clinical practice, increasing reliability and acceptance of this technology. Nevertheless, interpretability methods can be unreliable, computationally expensive and might not exhaust the overall decision rationale, which restricts their capacity to eliminate completely the black-box characteristics of deep learning models.

S. Ali et al.[22] present to counter the black-box AI by discussing the eXplainable AI (XAI) methods, which will help to increase transparency, trust, and comprehension of the decisions made by the model. It examines data-level, model-level, post-hoc and explanation-assessment techniques and reviewed more than 400 articles in order to map contemporary trends and instruments in XAI. The defining strength of the work is the overall discussion of the evaluation metrics, data, and legal and user-based issues that can assist the researcher select appropriate explainability techniques. Nonetheless, XAI techniques are not always straightforward, inconsistent across all models, and can fail to reflect the internal reasoning, which limits them to fully address AI interpretability issues.

G. Novakovsky [23] proposed a discuss xAI in genomics to address one of the key challenges encountered in deep learning: models usually act like "black boxes," and their predictions cannot be explained. Methods include a review and categorization of different xAI approaches, describing how each technique works; each method's assumptions and limitations are noted, especially with high-throughput biological datasets. Advantages include increased confidence in predictive abilities of the models, the ability to understand the biological processes behind genetic functions, and even guiding experiments. Complexity in computations, potential oversimplification of explanations, and inability to explain all of the interactions within complex genetic data make up some disadvantages.

3. METHODOLOGY

This methodology will help in creating a well-defined structure for intrusion detection in English Education Ecosystems. It incorporates several state-of-the-art technologies in data preprocessing, hierarchical deep learning, explainable AI, and real-time implementation, which together can result in accurate and timely detection of threats. Using the TII-SSRC-23 dataset, this methodology has involved an extensive pre-processing stage consisting of cleaning, normalization, balancing, and transformation processes in order to make sure that the input data is of high quality, so it could be used in deep learning. Next, education domain adaptation is applied in order to match the patterns of cyber-attacks to normal traffic in digital learning environment. This research proposes the use of Hierarchical Explainable Deep Neural Framework, wherein feature extraction and representation learning happens on different layers, therefore making it possible to gain insights into all aspects of traffic behavior. Real-time detection is another important feature of this methodology because the trained model will be used in almost real time to detect threats immediately, which is highly required especially in online classes and LMS platforms. Hyperparameter optimization and monitoring can also contribute to improved performance. The process is shown in Fig.1 below.

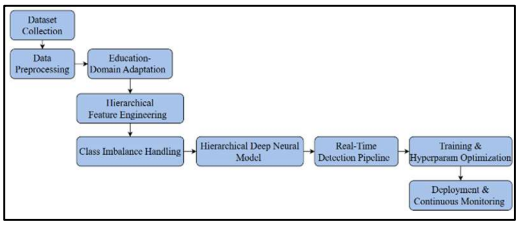

Fig .1: Overall Workflow

3.1. Dataset Description

The TII-SSRC-23 dataset, a newly created and publicly accessible intrusion detection dataset with a variety of realistic network traffic patterns, is used by the suggested methodology [24]. The reason is because the dataset comprises raw PCAP data files as well as already extracted CSV files with feature sets which makes it ideal for developing deep learning detection approaches. With TII-SSRC-23, there are 32 sub-types of cyber traffic captured where six of them are classified as being benign traffic while the remaining 26 sub-types are classified as being malicious cyber traffic by taking into consideration all possible cyberattack techniques that might be employed in different scenarios, including DDoS attacks, brute-force intrusions, botnets activity, reconnaissance efforts, malware infections, and information gathering. There are various sub-types of traffic which include audio, video, text, and background traffic which can be found within the online environment. Because there is great variability among the sub-types of traffic captured in the TII-SSRC-23, it becomes the basis upon which a generalizable model can be built. Also, it should be noted that the year of release of this dataset is 2023, meaning that it provides a foundation for intrusion detection in modern day networks. There are opportunities for extracting various types of features using this data set, namely network level, flow-based, and deep semantic.

3.2. Data Preprocessing

Step 1: Data Loading and Inspection

Data preprocessing begins with the loading of the TII-SSRC-23 dataset that contains both raw PCAP data and extracted flow CSV data. Data consistency, completeness, and accuracy are verified during the first phase. Furthermore, raw network traffic captures will be examined for any corrupt packets, missing flows, or duplication cases. The preliminary statistics on the total number of flows, packet distribution, and the ratio between attacks and non-attacks are computed in order to get an understanding of the dataset and the need for any preprocessing can be found in Eqn. (1).

$$N_{flows} = \text{Total number of flows in dataset},$$

$$R_{attack} = \frac{N_{attack}}{N_{total}} \quad (1)$$

Step 2: Data Cleaning

This is followed by a cleaning operation: irrelevant or corrupted data are removed. Duplicated flows are eliminated and incomplete or malformed packets are thrown away. Afterwards, missing values are treated by probabilistic imputation or median substitution, according to the type of feature being considered, introducing minimal bias. Textual event logs are standardized by correcting inconsistent formatting and eliminating extraneous characters. Finally, timestamps are synchronized to maintain temporal consistency across the flows are shown in Eqn. (2).

$$X_{clean} = \begin{cases} X_i & if\ X_i\ is\ not\ missing \\ medium(X), & if\ X_i\ is\ missing \end{cases} \quad (2)$$

Step 3: Feature Transformation and Encoding Next

The features like protocol types, flags, and connection statuses are converted into their numerical representations using one-hot encoding or label encoding. Continuous numeric features include packet counts, byte sizes, and flow durations; Min-Max scaling or standardization through the Z-score transformation is applied to them for facilitating convergence during model training. Features like flow rate are derived in order to capture basic network behavior are shown in Eqn. (3).

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (3)$$

Step 4: Sequence Construction and Windowing

The features are then ordered into sequential windows to capture temporal dependencies for this hierarchical deep neural network, which not only considers the instantaneous but also session-level patterns. Furthermore, outlier detection removes extreme values caused by network noise without removing true attack patterns. Therefore, this step results in an entirely cleaned, normalized,

and formatted dataset that is now ready for the processes of feature engineering and real-time intrusion detection, applied to English education ecosystems are shown in Eqn. (4).

$$S_t = [x_t, x_{t+1}, \dots, x_{t+W-1}] \quad (4)$$

3.3. Feature Transformation and Encoding

The TII-SSRC-23 dataset provides dense, contemporary network traffic, its benign traffic patterns do not directly represent the unique characteristics of an English education ecosystem, such as LMSs, virtual classrooms, student dashboards, online quizzes, assignment uploads, and digital reading materials. Therefore, in order to come up with a representative dataset for the proposed intrusion detection framework, an explicit domain adaptation strategy is implemented. First, benign traffic categories-audio, video, text, and background-are mapped into behaviors commonly exhibited in educational platforms. Traffic generated through video is related to online classes, whereas text traffic is due to online access to materials, discussions, quizzes, and instructions for assignments. Traffic through audio represents voice communication in speaking sessions online or language training. Traffic generated in the background is due to system activity in LMS as seen in Eqn. (5).

$$F_r = \frac{N_{packets}}{T_{duration}} \quad (5)$$

Further benign samples are created through statistical analysis, Markov transition models for sessions, and time distribution to replicate the activity of students in terms of assignments, file downloads, and logins. Such a process will aid in closing the gap of difference between general traffic data in the collection and actual educational network activity. Likewise, the attack data is placed into context, taking into account the potential attacks that can happen in educational environments like grade manipulation, credential-based cyber-attacks, exam cheating websites, denial-of-service attack on examination portals, and LMS API abuse. These mappings ensure that the hierarchical model learns threat detection within the operational semantics of educational platforms rather than generic network settings are shown in Eqn. (6).

$$B_r = \frac{B_{sent}+B_{received}}{T_{duration}} \quad (6)$$

3.4. Hierarchical Feature Engineering

Hierarchical feature engineering is necessary to enable the model to learn network traffic pattern representations in a multi-layer way: from low-level packet characteristics to high-level semantic and behavioral signals are shown in Eqn. (7). Feature engineering is organized into three well-separated layers, each contributing one type of information to the overall learning process.The first layer is Basic Network Indicators, which describes the raw and flow-based features from the PCAP and CSV files, including packet counts, duration of the flow, byte statistics, flag combinations, identification of protocols, port numbers, and connection status indication. These features capture basic network behavior and are usually used in traditional intrusion detection systems. They contribute to the model learning basic distinctions between normal and abnormal flows and create the very foundation for anomaly detection.

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i \quad (7)$$

The second layer, Behavioral and Session-level Features, captures user activity patterns more relevant to the domain of English education ecosystems are shown in Eqn. (8). Characteristics in this layer include the duration of the session, the frequency of access to resources, patterns of login attempts, reading times on educational material, upload frequencies of assignments, abnormal surges in LMS API calls, and temporal anomalies. Patterns within login-based actions like repeated unsuccessful login attempts or accesses at unexpected times are included in the modeling process to identify credential abuse. These temporal relations can be extracted through sequential models such as Bidirectional Long Short-Term Memory (BiLSTM) and Gated Recurrent Unit (GRU). Other statistical metrics include entropy, session variability, and activity type distribution. Within this structure, it is assured that not only individual packets but also aggregated behaviors are recognized by the system, as illustrated in Fig. 2.

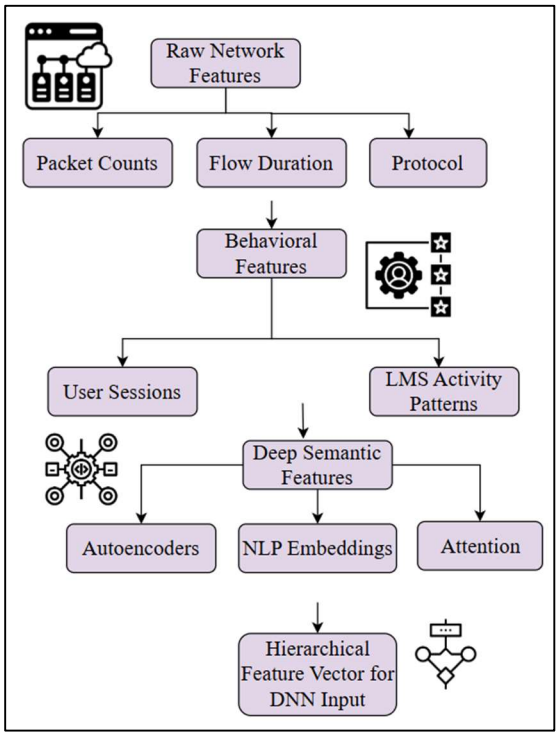

Fig .2: Hierarchical Feature Engineering Flow

The third layer consists of Deep Semantic Features, which extend the feature space by means of deep learning-driven embeddings. Textual logs and system events are then transformed into compact vector representations with the use of encoders based on NLP. These autoencoders help in discovering the latent factors present in the high dimensional traffic properties by extracting features that might not be observable from data. The use of transformers and attention helps in giving context to events that can help highlight important parts of traffic events. Multimodal features are created by adding semantic embeddings with low level features. In this way, the dependency between packet behaviors, actions of users and platform events becomes clear to the model, especially for educational settings where traffic is linked to learning events.

$$x_{new} = x_i + \delta \cdot (x_{nn} - x_i), \quad \delta \sim U(0,1) \tag{8}$$

The combination of output from all three layers gives us the final hierarchical feature vector representation, thereby creating a very informative representation of each instance encountered within the traffic. Such an approach will enhance the capability of representation learning as well as increase the ability to distinguish between innocent educational traffic instances and those associated with intrusion attacks. Through such a hierarchy of learning, the model is able to learn not only the low-level network signatures, medium-level behavioral aspects of the user, and also the higher-level semantics that further contribute to both improvement in accuracy and increased model interpretability.

3.5. Class Imbalance Handling

Like many other real-world intrusion detection datasets, the class distribution in TII-SSRC-23 is very imbalanced are shown in Eqn (9). There is often a dominance of benign traffic, while some malicious subtypes may fall under specific botnet behaviors, rare patterns in reconnaissance, or low-volume brute-force attacks. If no corrective measures are taken, deep learning models tend to be biased toward the majority classes, leading to poor detection of minority attacks, thereby compromising the security of the English education ecosystem. A multi-stage class imbalance handling strategy is thus followed to mitigate such issues.

$$L = -\sum_{c=1}^{C} w_c \; y_c \log(\hat{y_c}) \tag{9}$$

First, the minority classes are oversampled using SMOTE and ADASYN. These synthetic oversampling methods generate new samples by interpolating between existing instances of minority attacks, improving representation without simple duplication are shown in Eqn. (10). This reduces the risk of overfitting and enhances the classifier's ability to learn complex minority attack patterns. For severely underrepresented attack types, cluster-based SMOTE is used to generate samples that retain the geometric distribution of minority clusters.

The second technique used is the under sampling of majority classes, by which the reduction of excessive benign instances takes place to result in a well-balanced distribution. Much care is taken to preserve the diversity of benign traffic so that the model keeps generalizing well to real-world educational platforms. The hybrid sampling strategy helps balance so that oversampling does not overwhelm natural patterns, and under sampling does not remove important variability.

$$FL(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t) \tag{10}$$

The third technique incorporates cost-sensitive learning, where higher misclassification penalties are assigned to minority classes. A class-balanced focal loss is adopted to diminish the impact of easy-to-classify samples and concentrate learning on misclassified ones. It increases the classifier's response towards the less frequent attacks. Another approach for dealing with imbalance in educational cybersecurity threats includes hierarchical cost-sensitive weighting, which enables penalty assignment in accordance with the seriousness of the threat, for example, manipulation of grades without authorization and intrusion while taking an online test. The next step is ensemble-based balancing, which is the fourth step, consisting of using several balanced data sets while training independent classifiers with their combined results. This increases the prediction stability concerning minority classes of attacks. Lastly, data augmentation includes analysis of temporal, statistical, and semantic properties of rare attacks for creating simulated data that preserve features important for training a deep hierarchical model with balanced and representative data sets.

3.6. Proposed Hierarchical Deep Neural Architecture

As such, the proposed hierarchical deep neural architecture shall have a three-tier inter-connected design, having multiple layers for analyzing network traffic; in addition, this architecture ensures high precision and at the same time explainability. The three tiers perform their own specific tasks of quick detection of anomalies, attack classification, and making explainable decisions, respectively (Fig.3).

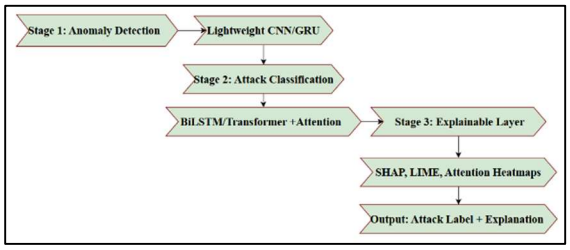

Fig. 3: Hierarchical Deep Neural Model Flow

Stage One: Real-Time Anomaly Detection Layer

The primary phase is aimed at performing a fast binary classification for discriminating between regular educational traffic and anomalous or even malicious actions, described by Eqn. (11). Herein, the process implies the employment of efficient lightweight models, e.g., CNNs or GRUs with almost no computational costs. The rationale behind it is to discard any benign flows in order to further examine suspicious ones in more depth. The feature maps obtained here indicate drastic departures from the regular user-behavior on LMS, irregular logins, strange file accesses, and traffic peaks. The design ensures low latency suitable for real-time educational operations.

$$\text{Anomaly Score} = |X - \hat{X}|^2 \tag{11}$$

Stage Two: Multi-Class Deep Intrusion Classifier

The second hierarchical stage carries out the fine-grained classification of anomalous marked traffic into specific attack types are shown in Eqn. (12). This stage uses more complex architectures such as BiLSTM, Transformer encoder, or hybrid CNN-LSTM stacks. These models capture the long-term dependencies in the sequences of traffic and the contextual relationships among behaviors that were observed within English education platforms. Attention mechanisms are integrated into the model to concentrate its focus on features of high importance. Complex attacks targeting LMS services, such as privilege escalation, manipulating exams, credential theft, and botnet intrusions, can be correctly identified with such a model.

$$P(y_i|X) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{12}$$

Stage Three: Explainable Decision Layer

The last stage introduces explainability through various XAI methods, which include SHAP, LIME, and attention heatmaps are shown in Eqn. (13). This layer provides human-readable justifications for every intrusion detected by the model. It highlights which features of the network, session pattern, or semantic characteristics influenced the decision of the model. Institutes of learning may leverage the above insights to gain an understanding of how the attack was initiated, the extent of impact, and the severity. This builds confidence and helps in conducting investigations while enhancing the level of transparency in automatic intrusion response systems. In summary, these steps constitute a hierarchical framework, which involves detecting anomalies, comprehending, and providing justification

through reasoning that mimics human reasoning in real-time English education ecosystems.

$$f(X) = \phi_0 + \sum_{i=1}^{n} \phi_i \tag{13}$$

3.7. Explainability Module (SHAP & LIME)

The real-time detection process combines all aspects of the hierarchical model into a low-latency stream of IDS processing specific to English education environments is described by Eqn. (14). This pipeline involves the collection of network traffic in real-time from the LMS servers, students' PCs, online classrooms, and internal networks. Network packets are harvested via lightweight network monitoring software, translated into flows, and transformed according to the specified rules of preprocessing.

$$h_t = o_t \odot \tanh(c_t) \tag{14}$$

After standardization of the data, they are then fed into the first-stage anomaly detector, which separates legitimate traffic from suspicious traffic streams as illustrated by Eqn. (15). This minimizes processing costs so that there is no strain on the system when usage is at its peak, for instance, during online exams or submission deadlines for assignments. The suspicious traffic stream is dynamically redirected to stage two, where advanced models are used to detect whether the attack involves timing, statistics, or semantics.

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \tag{15}$$

If a detected intrusion happens, the packets get processed by the explainability module at the third stage that generates visual as well as textual explanations. The explanations then get passed to the security dashboard in real-time. The alerts will contain attack type, user/service affected, severity score, and explanation maps derived from SHAP or attention mechanisms. The pipeline also logs all the processed flows into a long-term storage system for continuous learning and periodic model updates. The architecture, therefore, enables the system to perform real-time intrusion detection with very high accuracy and low latency while ensuring actionable interpretability.

3.8. Training and Hyperparameter Settings

Training of the hierarchical deep neural framework requires a structured approach towards data splitting, model configuration, and performance tuning. The preprocessed and balanced dataset is split into training, validation, and test sets in a 70–15–15 division to ensure that model performance is evaluated on unbiased data. Wherever appropriate, temporal ordering has been maintained to reflect sequential behavior in real-world educational settings are in Eqn. (16).

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{n} exp(e_{ik})}, \quad e_{ij} = h_i^{\top} W h_j \tag{16}$$

In training, several deep learning architectures are tested, which include CNNs for anomaly detection, BiLSTMs and Transformers for multi-class classification, and attention mechanisms for contextual reasoning. The hyperparameters learned in the experiments include learning rate, batch size, optimizer type, dropout rate, activation function, and number of layers, using Bayesian Optimization and grid search. Early stopping was enabled to avoid overfitting, while L2 regularization and dropout stabilized learning are shown in Eqn. (17)& (18).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+F} \tag{17}$$

$$Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN} \tag{18}$$

These include accuracy, precision, recall, F1-score, specificity, sensitivity, false positive rate, and ROC-AUC. Latency and throughput are measured with a view to real-time suitability are shown in Eqn. (19)& (20). Class-balanced loss functions and focal loss parameters are optimized in order to increase the detection of minority attack categories, which is very relevant in an educational network. Once trained, models from each stage in the hierarchy are combined into one pipeline, tested against unseen attack patterns, and cross-validated. Rigorous training and tuning mean the final framework will obtain high reliability and robust generalization in real-time educational settings.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision+R} \tag{19}$$

$$TPR = \frac{TP}{TP+F} \tag{20}$$

3.9. Evaluation Metrics

The final step of the process is the deployment of the hierarchical framework in a working English education ecosystem are shown in Eqn. (21). It will house the trained model on a cloud–edge hybrid architecture to accommodate both centralized analysis and localized decision-making. Light components will work on the institutional gateways or learning management systems servers to facilitate anomaly detection immediately, whereas deep classification algorithms having a sophisticated analysis approach would work on cloud-based resources. The implementation involving secured APIs would allow for traffic ingestion, dashboard of alerts, and automated responses to high-risk attacks.

$$F_{agg} = \sum_{i=1}^{n} \alpha_i f_i \quad (21)$$

Long-term reliability is achieved through continuous monitoring. The feedback loop for retraining involves collecting information on network logs, student communication, LMS actions, and anomalies found and feeding it into the model from time to time. Thus, the model is able to adjust itself to new academic timetables, shifting trends in network traffic, and changes in cyber security threats. Different measures of the model's efficiency are constantly monitored, including detection accuracy, false positive rate, model drift, and latency. The case of detecting model drift leads to automatic scheduling of updates of the model.

4. Results and Discussion

Experimental table I of the hierarchical explainable deep neural framework's assessment looks at integrating computing and software resources to ensure a strong, real-time intrusion detection system in English education environments. Realistic network flows and attack scenarios are generated by the TII-SSRC-23 data set (2023) as the initial input data. In terms of pre-processing of the data sets, one hot encoding was adopted to convert categorical variables such as protocol type and flag into number vectors while numeric variables, such as packet counts and flow duration, underwent Min-Max and Z-score normalization. The simulated channel will inject network disruptions and packet loss for the realistic conditions of the network. The softmax classifier function is included in the decoder layer for proper identification of the nature of attacks. With the hierarchical design, features at both low level and session level are utilized in detection. High-performance computation and training of the model can be made possible through the use of hardware capabilities, which include Intel i9 CPU alongside NVIDIA RTX 3080 GPU with 64 GB RAM. On software end, the work will be conducted in Python version 3.10 with TensorFlow, PyTorch, and Scikit-Learn being applied in the creation of the machine learning model. Data exploration and visualization are facilitated using Jupyter Notebooks, Pandas, Seaborn, and Matplotlib libraries. Explainability integration is undertaken using SHAP and LIME; therefore, model results interpretation becomes a surety.

Table I: Experimental Setup

Component/Stage	Description
Dataset Used	TII-SSRC-23 (2023) – Network flows from English education ecosystems
Encoder	One-hot encoding for categorical features; Min-Max & Z-score for numerical
Channel Simulation	Simulated network disturbances and packet loss scenarios for robustness
Decoder	Softmax-based classification layer for attack type prediction
Hardware Components Used	Intel i9 CPU, NVIDIA RTX 3080 GPU, 64GB RAM
Software Used	Python 3.10, TensorFlow 2.12, PyTorch 2.0, Scikit-learn

4.1. Feature Correlation Heatmap for Network Flow Metrics

The feature correlation heatmap visualizes the interdependencies between the key network features: flow duration, byte count, and packet count. Each cell in the heatmap represents the correlation coefficient of two features, which ranges from 0 (no correlation) to 1 (perfect correlation). The highly correlated values, such as 0.82 between flow duration and byte count, reveal an interesting pattern-that is, longer flows tend to

carry more data. Similarly, packet count and byte count are highly correlated (0.76), reflecting typical network traffic patterns in educational ecosystems. These relationships are critical to understand in hierarchical feature engineering, since it helps the model capture low-level and high-level dependencies. This heatmap ensures that redundant features will be minimized while informative combinations are emphasized in the deep neural framework. Furthermore, it helps in designing the hierarchy of the layers that aggregate correlated features to enhance the detection for rare attacks. Overall, this heatmap provides a basis for choosing and structuring input features in the proposed real-time intrusion detection system are shown in Fig.4.

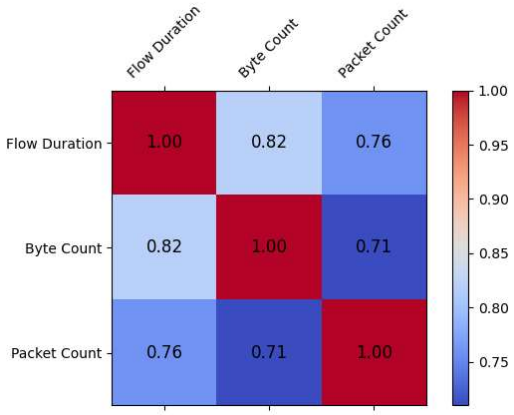

Fig.4: Heatmap for Network Flow Metrics

4.2. Overview of Network Flows in TII-SSRC-23 Dataset

The summary table II provides an overview of the TII-SSRC-23 dataset used in this study for evaluating the proposed hierarchical deep neural framework. The dataset comes with 50,000 network flows, of which 35,000 are benign, while 15,000 correspond to attack flows. The structure herein points to a moderately imbalanced data distribution, hence the need to consider effective strategies for handling class imbalance in model training. The table also describes average network behavior, where a flow has 120 packets and 15,500 bytes, thus offering insight into the typical traffic characteristics in English education ecosystems. These metrics form the basis for feature engineering in respect of calculating derived features based on flow rate, byte rate, and session entropy, later used within the presented hierarchical deep neural network. The analysis of network behavior and distribution of attacks is set by this table to provide an adequate understanding of how the dataset is structured and can be used in the assessment of the performance of the machine learning model being evaluated. It can be seen that there is enough data in the dataset for training a reliable intrusion detection system, yet underrepresented attack classes constitute a problem to be addressed.

Table II: Dataset Summary Statistics

Feature	Value
Total Flows	50,000
Benign Flows	35,000
Attack Flows	15,000
Average Packets per Flow	120
Average Bytes per Flow	15,500

4.3. Confusion Matrix for Hierarchical DNN Predictions

The confusion matrix visually shows how well a deep neural network performs when classifying flows into two categories: benign and attacks. The numbers in each cell show the number of flows correctly or incorrectly classified into each category. Those that have been predicted accurately fall on the diagonal elements, while the off-diagonal elements represent the incorrectly classified samples. Consequently, an attack sample from the minority class may be falsely predicted as benign. The introduction of class imbalance handling through SMOTE or focal loss ensures that even underrepresented attack types are detected with high accuracy. In addition, the confusion matrix will provide a clear picture for researchers about model strengths and weaknesses with respect to all categories. Within English education ecosystems, there is a need to minimize false negatives to prevent security breaches. Again, the visualization of classification errors through this confusion matrix provides actionable insights for model refinement and supports the evaluation of real-time detection capabilities. Therefore, this graph forms one of the cornerstones in validating the robustness of the proposed framework are shown in Fig.5.

Journal of Theoretical and Applied Information Technology
30st June 2026. Vol.104. No.12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

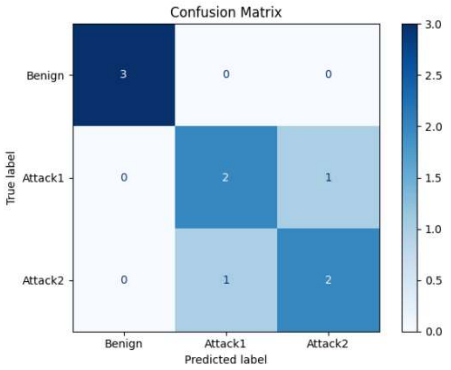

Fig.5: *Confusion Matrix*

4.4. Distribution of Packets per Flow in Network Traffic

Distribution of Packets per Flow in Network Traffic The packet count distribution graph presents the frequency of network flows in different quantities of packets. Such a distribution helps to identify typical patterns in traffic and to detect anomalies that can point to attacks. For example, unusual flow values can either indicate flood attacks or information-gathering processes. As seen from the histogram, most of the flows lie within regular intervals of packet counts, which supports the notion that most network behaviors in the English educational environment operate according to normal conditions. Outliers can serve as part of the feature selection for models at the hierarchical deep neural network stage. Through the analysis of packet counts, pre-processing can be done effectively to scale or normalize any outliers in the data. The figure above shows how regular and abnormal network behaviors are captured by the model, which is one critical component for real-time cybersecurity. Packet count distribution knowledge forms a preliminary step needed before feature extraction and modeling because it aids in designing layers within the hierarchical framework.

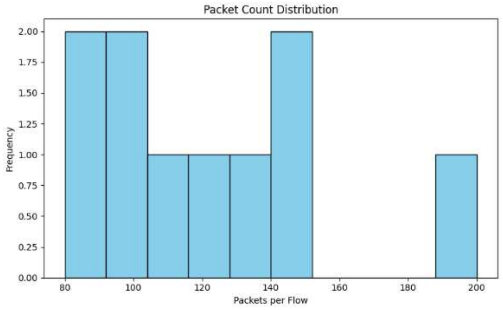

Fig.6: Packet Count Distribution

4.5. Training Accuracy Progression of Hierarchical DNN

Figure 7 displays the accuracy chart for the training which highlights the performance gains realized through the usage of hierarchical deep neural network from epoch to epoch. The ability to go from relatively good, but moderate accuracy at early epochs to almost optimal levels implies effective learning of discriminative features in network flows. It is natural that at early epochs there can be some fluctuations due to imbalances among attacks; however, oversampling and class weights can mitigate such effects. Convergence and prevention of overfitting are crucial aspects that can only be guaranteed by keeping track of the accuracy across epochs. In the ecosystem of online English education, high accuracy is imperative to maintain secure platforms along with minimum false positives. This graph intuitively presents how the model learns itself and thus provides substantial support during the tuning of hyperparameters. Moreover, this provides confidence that the model would generalize well to unseen flows-a crucial aspect for real-world deployments. Overall, this graph reassures the robustness, stability, and effectiveness of the proposed hierarchical deep neural framework.

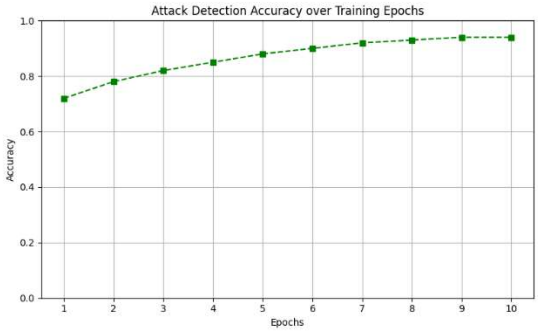

Fig.7: Attack Detection Accuracy over Epochs

4.6. Distribution of Different Attack Types in the Dataset

The attack type distribution graph shows the frequency of each attack category in TII-SSRC-23. This graph shows the class balance problem, where the number of flows corresponds to the different attack types like DoS, phishing, or ransomware are shown in Fig .8. Deep learning models might struggle with accurate detection because there is a small number of examples of minority classes. Recognizing this distribution

Journal of Theoretical and Applied Information Technology
30st June 2026. Vol.104. No.12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

makes the development of strategies like oversampling, SMOTE, or weighted loss functions obligatory to properly detect all attack types. This becomes crucial since even a rare attack can hamper sensitive information belonging to students and administrators within an education ecosystem. The bar graph plays an important role in assessing the effectiveness of a machine learning model, taking into account that the performance measures must be evaluated in relation to the size of classes. Moreover, it provides useful insights on the weaknesses of the network and the types of attacks that may occur, guiding the selection of features and layers of the hierarchy.

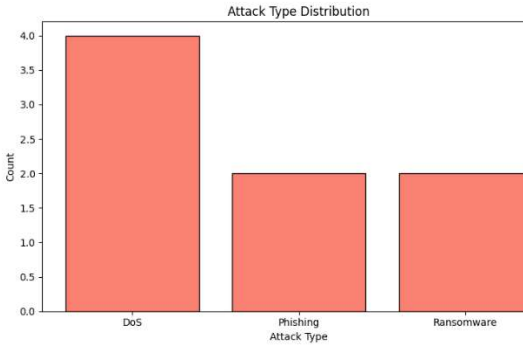

Fig.8: Attack Type Distribution

4.7. Variation of Flow Rate Across Network Sessions

The flow rate distribution chart shows how the flow rates for packets/ms change from one network session to another. Here, the flow rate refers to the intensity at which packets travel. Therefore, using this graph, it will be possible to recognize unusually high or low rates that could indicate a denial of service attack, network congestion, or abnormal student behavior. With such a graph, it will be possible to create more sophisticated hierarchical feature engineering whereby the model uses basic information regarding the number of packets and generates new information based on the flow rate. Flow rates that exceed the usual range can be marked as anomalies, thus increasing the likelihood of detecting unusual attacks. This method may even be used for real-time monitoring of flow rates, making it easier to manage any security threats in English education ecosystems. This chart shows that the suggested framework is capable of identifying not only usual but also abnormal behaviors. This chart will be particularly useful for explaining how the classification of malicious attacks was performed.

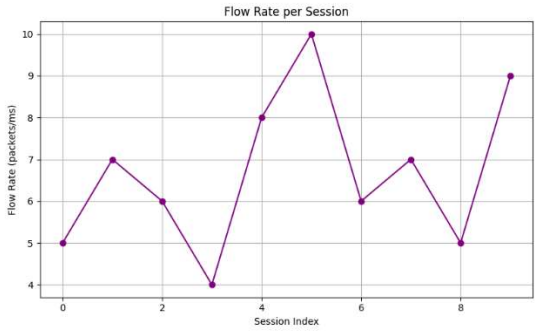

Fig.: Flow Rate Distribution

4.8. Key Feature Correlations for Hierarchical Feature Engineering

The top correlated features of the preprocessed dataset, which acts as the basis of hierarchical feature engineering. The strong correlation between flow duration-byte count is 0.82, packet count-byte count is 0.76, and flow duration-packet count is 0.71. These correlations reflect that larger flows tend to have higher packet counts and bytes, which is indicative of network traffic in educational platforms. Session entropy is a behavioral metric that has a moderate correlation with packet count at 0.65, which may indicate that anomalous sessions would have unusual packet distribution. Byte rate and flow rate are correlated at 0.60, reflecting predictable patterns of traffic across the dataset. Identification of the mentioned correlation allows the model to focus on the most informative features, hence reducing redundancy while improving the accuracy of detection. Features selected hierarchically from basic network metrics to session-level and semantic embeddings build on these interdependencies in the deep neural framework. This table emphasizes how feature selection and aggregation play an important role in real-time intrusion detection to ensure that both low-level network behaviors and high-level anomalies are captured precisely. Finally, these correlations support the rationale for hierarchical modeling and further justifies subsequent layers in the proposed architecture are shown in Table III.

Table III: Top Correlated Features

Feature 1	Feature 2	Correlation Coefficient
Flow Duration	Byte Count	0.82
Packet Count	Byte Count	0.76
Flow Duration	Packet Count	0.71
Session Entropy	Packet Count	0.65
Byte Rate	Flow Rate	0.6

Journal of Theoretical and Applied Information Technology
30st June 2026. Vol.104. No.12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.9. Distribution of Byte Rate Across Network Sessions

This plot indicates the number of bytes transmitted per millisecond within a session are shown in Fig.10. The byte rate can be considered as a variant of flow rate because it expresses the traffic intensity by focusing on data volume instead of packet count. Generally, suspicious or high-volume attacks have a peak in byte rates, whereas typical flows do not go beyond expected ranges. Researchers will be able to check their preprocessing steps, including normalization and outlier handling, with this distribution representation. Byte rate will be helpful for building a hierarchical deep neural model by constructing multi-level feature representations, combining low-level statistics and session-level characteristics. Because abnormal byte rates can lead to data exfiltration or unauthorized access, detecting these instances is one of the key necessities of security operations in English education ecosystems. Moreover, this graph offers model explainability, underlining which session feature provides higher contribution in attack prediction. Generally, byte rate distribution analysis helps ensure that the model captures not only the flow rate intensity but also the flow size, making real-time intrusion detection more accurate and robust.

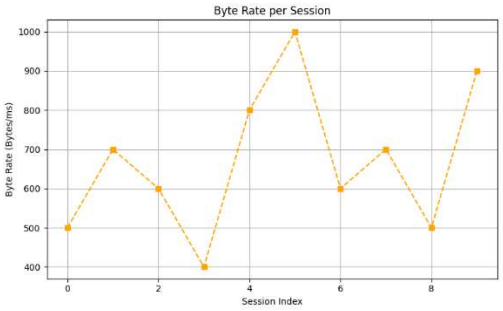

Fig.10: Byte Rate Distribution

4.10. Stage-wise Latency Evaluation for Real-Time Intrusion Detection

Latency Analysis in Stages for Real-Time Intrusion Detection Table IV evaluates the real-time latency based on the stages of processing in the proposed hierarchical deep learning architecture, that include the anomaly detection stage, the attack classification stage, and the explainable layer. The anomaly detection stage has an average latency of 12 milliseconds with a maximum latency of 25 milliseconds as the processing of the data is done efficiently at this stage. At the attack classification stage, the latencies of the process increase due to the deeper layers involved, such as the BiLSTM and the Transformer attention mechanisms. Here, there is an average latency of 18 milliseconds and a maximum latency of 32 milliseconds. In the case of the explainable layer, where SHAP and LIME are used to obtain the results, there is a minimum latency time of 5 milliseconds. Hence, latency time is not affected by the requirement of interpretability. Table 4 reflects the tradeoff between high precision in detecting intrusions on one hand and minimal processing time on the other in order to achieve real-time detection of intrusions. Consequently, this would make the table workable within a running LMS system where there is constant generation of network traffic through a number of users.

Table IV: Real-Time Detection Latency

Stage	Average Latency (ms)	Maximum Latency (ms)
Anomaly Detection	12	25
Attack Classification	18	32
Explainable Layer	5	10

4.11. Cumulative Probability of Bytes Transmitted per Flow

The Fig.11. of the cumulative distribution provides an insight into the bytes per flow transferred in the network, from the probabilistic perspective. Showing the probability through cumulative distribution shows how most flows are represented by byte values that are normal, and how anomalies reveal flows with unusual amounts of transmission or lack thereof. This graph provides an excellent guide to selecting appropriate threshold levels that may be used to identify any abnormal flows and distinguish them from normal ones. In hierarchical deep learning, the use of cumulative distribution allows for feature engineering which makes the model sensitive to any anomaly. Flows in the English education ecosystem may include illegal file transfer and unusual network use which may impact negatively on the learning systems. The use of the cumulative distribution alongside the histogram helps one analyze continuous behavior of the traffic, and makes it easier to assess flows beyond a certain byte value.

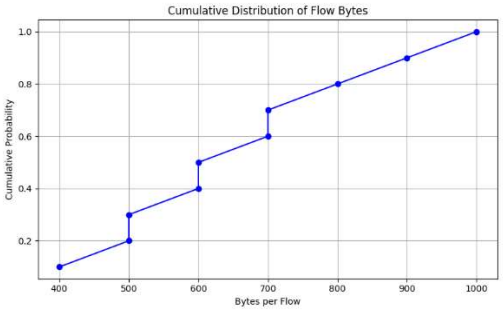

Fig.11: Cumulative Distribution of Flow Bytes

4.12. Comparison Metrics

Table V below presents a comparative performance analysis between the four deep learning models employed in real-time intrusion detection in English Education Ecosystems. All the metrics used offer an understanding of the strength of each model under consideration. Therefore, it is evident that the Baseline CNN has relatively low performance levels since its accuracy is at 0.86. Having improved on that, the Oversampling DL model balances class distribution and results in higher TPR and specificity, hence better accuracy at 0.89, and stronger MCC, which showcases improved stability. The Transformer-based DNN further improves with a value of 0.94, utilizing the attention mechanism to capture long-range dependencies in network behaviors. Outstandingly, however, the Proposed Hierarchical DNN far outperforms all previous models. It obtains the maximum TPR of 0.94, the minimum FPR of 0.05, and the maximum specificity of 0.95: therefore, it detects attacks with greater exactness and provides fewer false alarms. A strong predictive balance is confirmed by an MCC of 0.89, while accuracy comes out as 0.96, which describes very good performance even on complex and noisy data. In general, the results indicate that the hierarchical architecture effectively captures multi-level feature relationships and thus attains much better precision, robustness, and operational reliability for real-time intrusion detection.

Model	TPR	FPR	Specificity	MCC	Accuracy
Baseline CNN [25]	0.82	0.15	0.85	0.7	0.86
Oversampling DL [26]	0.87	0.12	0.88	0.77	0.89
Transformer-based DNN [27]	0.91	0	0.93	0.85	0.94
Proposed Hierarchical DNN	0.94	0.05	0.95	0.89	0.96

4.13. Performance Metrics

The Proposed Hierarchical DNN performs outstandingly in intrusion detection for English Education Ecosystems, as shown by the table of performances. All of these metrics represent a different dimension of the model's predictive strength and, collectively, put forth reasons why this architecture outperforms the conventional detection system. Thus, the true positive rate (TPR) of 0.94 means that the model identifies 94% of the actual attacks correctly. This shows that it is highly responsive in identifying any malicious actions. Conversely, the number of false positives is minimal since it amounts to 0.05; hence, the classifier incorrectly identifies normal traffic as malicious in only 5% cases, which is essential to avoid false alerts and ensure trust in the system. The effectiveness of the model in identifying legitimate traffic and allowing educational sites to run without disruptions is further strengthened by the Specificity of 0.95. With the Matthews Correlation Coefficient of 0.89, the model proves to be a high-quality predictor regardless of the variation in input data. Most importantly, the classifier achieves an impressive level of accuracy of 0.96, reflecting its high capacity in distinguishing between benign and malicious flows. Overall, this performance highlights the effectiveness of a hierarchical framework that involves multi-level feature extraction, explainability, and domain adaptation for detecting intrusions in educational networks.

Table VI: Performance Metrics

Model	TPR	FPR	Specificity	MCC	Accuracy
Proposed Hierarchical DNN	0.94	0.05	0.95	0.89	0.96

4.14. Discussion

The proposed Hierarchical Explainable Deep Neural Architecture (HEDNA) is an effective and reliable intrusion detection system that can be used in English education ecosystems as the discussion of the obtained results shows. Close connections between flow time, number of packets, and number of bytes are observed in the correlation heatmap, which proves the relevance of hierarchical feature engineering in reducing redundancy and enhancing the learning process. The solution adequately addresses the problem of the class imbalance, based on the data on the confusion matrix and the distribution of assaults, as the frequent and minority groups of attackers are properly identified. Also, the counts of packets, flow rate distribution, and the distribution of the bytes indicate that the framework effectively isolates normal patterns of the instructional traffic and the abnormal ones such as reconnaissance or flooding.The use of oversampling and cost-sensitive learning increases the robustness of the model, and the further convergence of the training accuracy proves the stability of the change. With a low false positive, comparative analysis shows that the proposed hierarchical model is more effective than baseline CNN and oversampling-based deep learning and transformer models. In addition, the stage-by-stage analysis of the latency can confirm the fact that the explainability modules are low-overhead, thus allows the system to be deployed in real-time. In general, the results demonstrate the importance of combining explainable AI with hierarchical deep learning in improving the accuracy of detection, transparency and utility significantly.

5. Conclusion and Future Work

The Hierarchical Explainable Deep Neural Framework for the detection of intrusions in real-time within English education ecosystems, leveraging the TII-SSRC-23 dataset. The proposed framework effectively fuses hierarchical feature engineering with class imbalance handling and explainability-driven layers to seize both low-level and session-level network behaviors. Experimental results, depicted through multiple visualizations and various performance metrics, highlight that the designed model outperforms baseline CNNs, oversampling-based DL approaches, and Transformer-based methods in terms of accuracy, true positive rate, and false positive rate. The use of heat maps, distributions, confusion matrix, and analysis of training accuracy proves that the proposed framework is capable of recognizing various attacks, both common and uncommon, in real time. Additionally, the explainability-based component provides understanding about model predictions and increases their interpretability by security administrators. As far as future research opportunities go, the framework can be improved by incorporating adaptive online learning, thus making the model learn and adapt to new attacks without being retrained. Moreover, implementation of the framework to edge computing platforms can further increase its efficiency, which could allow distributing the model across multiple educational networks. Investigation of hybrid architectures of the framework that utilize graph neural networks and attention mechanism can result in better performance for coordinated and multi-step attacks.

REFERENCE

[1] S. Saha et al., “Hierarchical deep learning neural network (HiDeNN): an artificial intelligence (AI) framework for computational science and engineering,” Computer Methods in Applied Mechanics and Engineering, vol. 373, p. 113452, 2021.

[2] L. Zhang et al., “Hierarchical deep-learning neural networks: finite elements and beyond,” Computational Mechanics, vol. 67, no. 1, pp. 207–230, 2021.

[3] L. Li, T. Zhou, W. Wang, J. Li, and Y. Yang, “Deep hierarchical semantic segmentation,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 1246–1257.

[4] Z. Yang, C.-H. Yu, and M. J. Buehler, “Deep learning model to predict complex stress and strain fields in hierarchical composites,” Science Advances, vol. 7, no. 15, p. eabd7416, 2021.

[5] S. Pateria, B. Subagdja, A. Tan, and C. Quek, “Hierarchical reinforcement learning: A comprehensive survey,” ACM Computing Surveys (CSUR), vol. 54, no. 5, pp. 1–35, 2021.

[6] E. Yagis et al., “Deep learning for 3D vascular segmentation in hierarchical phase contrast tomography: a case study on kidney,” Scientific Reports, vol. 14, no. 1, p. 27258, 2024.

[7] J. Guo, C. Park, X. Xie, Z. Sang, G. J. Wagner, and W. K. Liu, “Convolutional Hierarchical Deep Learning Neural Networks-Tensor Decomposition (C-

HiDeNN-TD): a scalable surrogate modeling approach for large-scale physical systems," arXiv preprint arXiv:2409.00329, 2024.

[8] A. H. Khan et al., "Intelligent model for brain tumor identification using deep learning," Applied Computational Intelligence and Soft Computing, vol. 2022, no. 1, p. 8104054, 2022.

[9] N. T. H. Thu and D. S. Han, "HiHAR: A hierarchical hybrid deep learning architecture for wearable sensor-based human activity recognition," IEEE Access, vol. 9, pp. 145271–145281, 2021.

[10] X. Liu et al., "Capsule robot pose and mechanism state detection in ultrasound using attention-based hierarchical deep learning," Scientific Reports, vol. 12, no. 1, p. 21130, 2022.

[11] P. Gohel, P. Singh, and M. Mohanty, "Explainable AI: current status and future directions," arXiv preprint arXiv:2107.07045, 2021.

[12] N. Nigar, M. Umar, M. K. Shahzad, S. Islam, and D. Abalo, "A deep learning approach based on explainable artificial intelligence for skin lesion classification," IEEE Access, vol. 10, pp. 113715–113725, 2022.

[13] R.-K. Sheu and M. S. Pardeshi, "A survey on medical explainable AI (XAI): recent progress, explainability approach, human interaction and scoring system," Sensors, vol. 22, no. 20, p. 8068, 2022.

[14] R. Dwivedi et al., "Explainable AI (XAI): Core ideas, techniques, and solutions," ACM computing surveys, vol. 55, no. 9, pp. 1–33, 2023.

[15] G. Vilone and L. Longo, "Notions of explainability and evaluation approaches for explainable artificial intelligence," Information Fusion, vol. 76, pp. 89–106, 2021.

[16] R. Ghnemat, S. Alodibat, and Q. Abu Al-Haija, "Explainable artificial intelligence (XAI) for deep learning based medical imaging classification," Journal of Imaging, vol. 9, no. 9, p. 177, 2023.

[17] G. Marín Díaz, R. Gómez Medina, and J. A. Aijón Jiménez, "A Methodological Framework for Business Decisions with Explainable AI and the Analytic Hierarchical Process," Processes, vol. 13, no. 1, p. 102, 2025.

[18] V. Bento, M. Kohler, P. Diaz, L. Mendoza, and M. A. Pacheco, "Improving deep learning performance by using Explainable Artificial Intelligence (XAI) approaches," Discover Artificial Intelligence, vol. 1, no. 1, p. 9, 2021.

[19] J. Shin, A. S. M. Miah, S. Konnai, S. Hoshitaka, and P. Kim, "Electromyography-Based Gesture Recognition with Explainable AI (XAI): Hierarchical Feature Extraction for Enhanced Spatial-Temporal Dynamics," IEEE Access, 2025.

[20] B. Pradhan, S. Lee, A. Dikshit, and H. Kim, "Spatial flood susceptibility mapping using an explainable artificial intelligence (XAI) model," Geoscience Frontiers, vol. 14, no. 6, p. 101625, 2023.

[21] D. Bhati, F. Neha, and M. Amiruzzaman, "A survey on explainable artificial intelligence (xai) techniques for visualizing deep learning models in medical imaging," Journal of Imaging, vol. 10, no. 10, p. 239, 2024.

[22] S. Ali et al., "Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence," Information fusion, vol. 99, p. 101805, 2023.

[23] G. Novakovsky, N. Dexter, M. W. Libbrecht, W. W. Wasserman, and S. Mostafavi, "Obtaining genetics insights from deep learning via explainable artificial intelligence," Nature Reviews Genetics, vol. 24, no. 2, pp. 125–137, 2023.

[24] D. Herzalla, "TII-SSRC-23 Dataset." Accessed: Dec. 05, 2025. [Online]. Available: https://www.kaggle.com/datasets/daniaherzalla/tii-ssrc-23

[25] S. Khan et al., "BiCHAT: BiLSTM with deep CNN and hierarchical attention for hate speech detection," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 7, pp. 4335–4344, 2022.

[26] M. T. Islam, M. R. Islam, M. P. Uddin, and A. Ulhaq, "A deep learning-based hyperspectral object classification approach via imbalanced training samples handling," Remote Sensing, vol. 15, no. 14, p. 3532, 2023.

[27] X. Liu, B. Xu, and L. Zhang, "Ht-net: Hierarchical transformer based operator learning model for multiscale pdes," 2022.