National College of Ireland

Integrating SHAP and Attention Mechanisms for Transparent Deep Autoencoder Anomaly Detection

MSc Research Project
Msc in Data Analytics (MSCDAD_C)

Venkateshwar Rao Telu
Student ID: x23318392

School of Computing
National College of Ireland

Supervisor: Jaswinder Singh

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Venkateshwar Rao Telu
Student ID:	x23318392
Programme:	MSC in Data Analytics **Year:** 2024 - 2025
Module:	Research Practicum part - 2
Supervisor:	Jaswinder Singh
Submission Due Date:	September 15, 2025
Project Title:	Integrating SHAP and Attention Mechanisms for Transparent Deep Autoencoder Anomaly Detection
Word Count:	6589 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Venkateshwar Rao Telu

Date: September 15, 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Integrating SHAP and Attention Mechanisms for Transparent Deep Autoencoder Anomaly Detection

Venkateshwar Rao Telu

x23318392

Abstract

This study aims to develop and evaluate a transparent deep autoencoder anomaly detection framework by integrating SHAP-based feature attribution and attention mechanisms, thereby enhancing interpretability while maintaining high detection performance in network traffic contexts. This study follows the CRISP-DM framework, establishing a correspondence between the phases and the anomaly detection pipeline. The Network Anomaly Dataset has been collected in the form of a CSV on the Kaggle site, and the data has been loaded into Pandas using read into CSV. The adaptive combination of the information latencies across the channels and along time axes was realized by the interleaving blocks of CBAM in consecutive convolutional layers, and the more basic latent representations were realized by MultiHeadAttention. Additionally, perfectly tuned Isolation Forest demonstrated an extremely elevated level of outlier scoring of sparse anomalies in fewer splits, which functioned as a good benchmark. Post-hoc SHAP interpretations identified that session state, destination host metrics, and connection flag patterns were the key factors influencing anomalous reconstruction, and most auxiliary features were minor contributors.

Keywords: SHAP-based feature attribution, Network Anomaly, CBAM, Autoencoder, and Isolation Forest.

1. Introduction

Network infrastructures are increasingly confronted with advanced, high-volume attacks that can bypass customary signature-based protection. Anomalous patterns, ranging from covert lateral movement to distributed denial-of-service attacks, tend to appear subtly in high-dimensional traffic logs, and hence, early detection becomes both imperative and challenging (Diro et al., 2024). Similarly, the average cost per data breach amounted to about $4.9 million in 2024, an increase of 15% over the last three years and highlighting the dire economic consequences of missed or delayed detections (IBM, 2024). However, in the face of increasing threats, the global network security market is expected to grow at a compound annual growth rate (CAGR) of approximately 7.2% from 2024 to 2029 to a total market volume of approximately $111.0 billion by 2029 (Markets and Markets, 2024).

Network anomaly detection is essential to the protection of digital infrastructure in recent times. Current deep autoencoders have high detection performance but are not transparent enough, constraining trust and actionable results (Bouman and Heskes, 2025). The difficulty is to create a new interpretable anomaly detection model that not only detects malicious patterns in network traffic but also provides an explanation for why inputs are detected. This study bridges the gap by combining self-attention mechanisms and SHAP explanations within a deep autoencoder to emphasize expressive features informing decisions related to network anomalies.

This study aims to develop and evaluate a transparent deep autoencoder anomaly detection framework by integrating SHAP-based feature attribution and attention mechanisms, thereby enhancing interpretability while maintaining high detection performance in network traffic contexts.

The objective of the research are

- To design a deep autoencoder architecture augmented with attention layers for dynamic feature weighting in anomaly detection.
- To integrate SHAP-based explainability into the autoencoder framework for both local and global feature attribution.
- To implement the proposed model using Python, TensorFlow and the SHAP library, benchmarking it against baseline autoencoders.
- To evaluate detection performance and interpretability through quantitative metrics (e.g., AUC, F1-score) and qualitative analyses (e.g., SHAP plots, attention heatmaps).

The research question is

- How does combining SHAP-based explainability with attention mechanisms increase the transparency and efficiency of deep autoencoder models in network anomaly detection within Python-based systems.

The hypothesis of the research is

Alternate Hypothesis (H_1): Integrating SHAP-based feature attribution and attention mechanisms within the autoencoder will significantly reduce the false positive rate in anomaly detection compared to a standard deep autoencoder.

This study fills an essential gap in transparency for automated network anomaly detection and makes several important contributions. Firstly, by combining self-attention mechanisms with a deep autoencoder, it enhances representation learning by dynamically concentrating on the most valuable traffic features, resulting in stronger detection of sophisticated, changing cybersecurity threats (Wang et al., 2024). Additionally, the use of SHAP values offers strong,

game-theoretic justifications for every anomaly choice, allowing analysts to be alerted to individual features.
The present research assumes that the Network Anomaly records are clean, properly annotated, and that it is representative enough of the actual traffic information of both normal and attack categories. It assumes that categorical encoding and MinMax scales maintain important relationships among features and do not cause bias. It is believed that CBAM-based attention blocks and MultiHeadAttention layers help the autoencoder prioritize salient features when performing reconstruction.
The model is evaluated against one unchanging data that constrains its application to other network categories/scenarios and/or dynamic traffic patterns. The autoencoder may apply bias to the predominant anomalies due to class imbalance in the type of attacks. Modular items, such as CBAM, multiple heads of attention, and SHAP explainers, impose limitations on their use in high-throughput environments due to their significant computational costs.

2. Related Work

2.1 Introduction

The section critically reviews foundational research studies in anomaly detection, including attention-enhanced deep learning, deep autoencoders, and general machine learning methods. SHAP-based explainability is then investigated, the theoretical foundations developed, gaps in the literature determined, and a conceptual framework developed, preparing the ground for further analysis and synthesis.

2.2 Critical Analysis of Previous Research Studies

2.2.1 Importance of attention mechanisms in Deep learning models for anomaly detection

Attention mechanisms are significantly utilized in deep learning techniques, especially for anomaly detection in recent times. Thus, Hu et al. (2023) have the explicit goal of improving network traffic anomaly detection by incorporating a self-attention mechanism within an autoencoder based on LSTM, with the focus being on enhanced feature weighting and context sensitivity of cybersecurity. However, Landauer et al. (2023) review deep learning for log-data anomaly detection, but neither of them primarily deals with attention, rather listing architectures, and preprocessing techniques. Consequently, Darban et al. (2024) present a time-series anomaly taxonomy, identifying transformer-based models (which depend on multi-head self-attention) as a category in addition to CNNs and RNNs. Thus, whereas Hu et al. (2023) aimed to assess the standalone value of attention, the surveys place attention in the context of larger architectural trends.
Methodologically, Hu et al. (2023) build a custom UNSW-NB15-derived dataset, use LSTM layers to encode temporal dependencies, and layer self-attention to weight sequence features before reconstruction. Conversely, Landauer et al. (2023) borrow systematic literature review procedures, database querying, inclusion/exclusion criteria, and qualitative synthesis to chart approaches to anomaly detection but do not evaluate particular attention models. Darban et al. (2024) similarly conduct a structured review, categorizing models by forecasting, reconstruction, and transformer architectures, and outline how self-attention mechanisms underlie advanced representation learning in time series. The direct implementation of attention mechanisms contrasts with the meta-analytic methods of the two surveys.
Hu et al. (2023) described how inserting self-attention enhances detection accuracy and accelerates convergence but have limited generalizability outside their carefully crafted dataset and absence of interpretability metrics. Whereas, Landauer et al. (2023) identified a

research gap, such as most log-based deep models are neither explainable nor support incremental learning, effectively calling for attention-driven interpretability. However, Darban et al. (2024) pointed out the robust anomaly detection capability of transformers but caution that their high computational expense and limited testing on multivariate datasets. Together, these studies highlight the promise of attention mechanisms but expose requirements for wider benchmarking, interpretability testing, and real-world assessment.

2.2.2 Utilization of Deep Autoencoders for Anomaly Detection

Deep autoencoders help in learning the compressed representations of data and reconstructing input data to detect anomalies. Thus, Azzalini et al. (2025) seek to empirically compare several deep autoencoder architectures, namely plain, variational, convolutional, recurrent, and attention-enhanced variants, to detect anomalies in building electricity usage, treating environmental and operational inefficiencies. On the other hand, Bionda, Frittoli, and Boracchi (2022) intend to detect anomalous regions in textured industrial images through a Complex Wavelet Structural Similarity (CW-SSIM) loss-trained convolutional autoencoder for increased sensitivity to anomaly pattern departures. However, Tien et al. (2021) suggest an IoT-centric framework integrating supervised device-type classification with autoencoder feature learning and transfer learning to enable the automation of anomaly detection on diverse networked devices.

Methodologically, Azzalini et al. (2025) build electricity datasets from actual office buildings, train eight different autoencoder models, and compare reconstruction error and probability scores when introducing attention layers. Whereas Bionda, Frittoli, and Boracchi (2022) break down grayscale image patches into steerable-filter sub bands, train a symmetric CNN autoencoder with CW-SSIM as training loss and anomaly score, and combine overlapping patch reconstructions. Comparatively, Tien et al. (2021) initially categorized types of IoT devices through supervised neural networks and XGBoost feature selection, then used autoencoders to perform unsupervised dimension reduction and transfer learned models across edge sites, emulating attacks with malformed Modbus TCP requests.

Results indicate the variants used by Azzalini et al. (2025), which introduced attention to improve detection accuracy (0.72 precision in CNN forecast) and convergence speed while demanding new self-attention architectures to prevent potentially generalizing only poorly to electricity data. However, Bionda, Frittoli, and Boracchi (2022) have found that CW-SSIM training outperforms (AUC 0.977) MSE and SSIM losses for textured image anomaly segmentation, although their approach still only processes grayscale inputs and is not good at color anomalies. Conversely, Tien et al. (2021) showed that autoencoders perform better than one-class SVM and isolation forests (up to F1 98.36%) and facilitate successful transfer learning, although extended verification on a wide variety of IoT devices and attacks is necessary. Thus, these highlight the domain generality of deep autoencoders while emphasizing the need for specialized loss functions, architecture adaptations, and extended dataset testing.

2.2.3 Impact of Machine Learning and Deep Learning in Anomaly Detection

The application of machine learning (ML) and deep learning (DL) for anomaly detection has been widely examined, presenting different purposes, approaches, main contributions, and limitations in key pioneer studies. Al-amri et al. (2021) aimed to extensively map ML and DL methods for IoT data streams, defining network anomaly types, learning modes, and evaluation criteria to guide future research directions. Comparatively, Devineni, Kathiriya, and Shende (2023) seek to analyze ML-driven anomaly detection in private and government sectors, with emphasis on security, data integrity, and ethics in supervised, unsupervised, and semi-supervised methods. However, Rafique et al. (2024) are aimed at existing IoT network

anomaly detection trends, examining recent ML/DL frameworks, datasets, and deployment issues to inform scalable IDS (Intrusion Detection System) design.

Al-amri et al. (2021) used a systematic literature review, combining state-of-the-art algorithms into a taxonomy of data nature, anomaly types, window models, and performance measures. Devineni, Kathiriya, and Shende (2023) used descriptive case studies and algorithm summaries, showing the use of decision trees, SVMs, Random Forests, and neural networks on real-world datasets, and model trade-offs. Rafique et al. (2024) undertake a systematic survey, classifying works by ML vs DL methods, input modalities (e.g., time-series, images), and dataset properties, and identify gaps for real-time and resource-poor environments.

Al-amri et al. (2021) stated that ensemble and DL approaches (e.g., autoencoders) can reach 95% accuracy and low false positives, but remark on the lack of real-time benchmarks and management of streaming data that anomaly changes. Conversely, Devineni, Kathiriya, and Shende (2023) discovered that Random Forest achieves 99.4% accuracy in security contexts but warn that results might not be generalizable outside chosen industries and do not offer cross-model comparisons. Similarly, Rafique et al. (2024) saw transformer-based and 1D-CNN methods attaining F1 scores of approximately 0.97, but constraints in scalability, heterogeneous dataset assessment, and explainability of the analysis are missing. Thus, this comparative analysis underscores the imperative for unified benchmarking, evolving-stream validation, and transparency-driven evaluation in anomaly detection research.

2.2.4 Explainability in Deep Anomaly Detection Using SHAP

This section reviews key studies integrating SHAP for explainable deep anomaly detection, comparing their aims, methodologies, findings, and limitations. Alam et al. (2024) suggest the SXAD (Shapely eXplainable Anomaly Detection) framework to transform black-box log anomaly models into white-box systems by combining Kernel SHAP with DT, RF, and GB classifiers on HDFS logs, with accuracies of 99.99%, 99.85%, and 99.99%, respectively. However, Zou and Petrosian (2020) seek to deploy Shapley value explanations to both Decision Tree and DeepLog, considering individual events and event sequences as “players” to assign network anomaly decisions. Conversely, Wu et al. (2021) are concerned with video anomaly detection, mixing a denoising autoencoder with SHAP on high-level CNN features to interpret frame-level anomalies.

Method-wise, Alam et al. (2024) used post-hoc Kernel SHAP on ensemble classifiers learned over parsed HDFS logs, applying Drain parsing and TF-IDF features. Comparatively, Zou and Petrosian (2020) created a bi-level Shapley computation by initially considering one log event as players and then pairs of events to control coalition explosion, on top of Decision Tree and DeepLog models. Otherwise, Wu et al. (2021) captured contextual video features through pre-trained CNNs (PFPN, ResNet-101, JDE) and a denoising autoencoder (DAE), passed them through a denoising autoencoder, and used SHAP to explain abnormal keyframes.

Alam et al. (2024) discovered that SXAD enhances interpretability and achieves near-perfect detection rates, although it is restricted to HDFS log patterns and has no assessment on real-time streams. Whereas Zou and Petrosian (2020) showed substantial attribution of log events and sequences but lack qualitative evaluation ratings for explanation fidelity and suffer from combinatorial scaling problems. However, Wu et al. (2021) obtained 85.9% AUC on UCSD Ped1 and 92.4% AUC on Ped2 in seconds of training of DAE, but over-reliance on pre-trained visual models could constrain generalization across varied surveillance settings. Overall, this comparative study highlights the increasing significance of SHAP-driven transparency and emphasizes the requirement for more comprehensive benchmarking and scalability testing.

2.5 Theoretical Underpinning

The ***Information Bottleneck (IB) Principle*** is selected for this study as it encapsulates the process by which a model compresses input data while retaining only the most task-relevant information. The IB principle helps in understanding how neural networks learn and represent information to train dynamics and explore distinct phases of compression (Tishby, Pereira and Bialek, 1999; Tezuka and Namekawa, 2021). In this architecture, the bottleneck of the deep autoencoder imposes compression of high-dimensional network feature spaces, and self-attention layers further select those compressed representations most predictive of anomalies. SHAP then estimates the contribution of each feature to the anomaly score, how relevant it is under the Information Bottleneck trade-off. Collectively, this theory guarantees the model distills and discloses the essential information informing anomaly detection.

2.6 Literature Gap

Despite progress in SHAP-based explainability and attention-augmented autoencoders, current studies remain isolated as attention-based works have no solid interpretability metrics. Additionally, deep autoencoder assessments rarely incorporate feature-level explanations, and ML/DL reviews report high detection scores but highlight inconsistent real-time and changing-stream benchmarks. Moreover, SHAP usage is plagued by scalability and domain-generalization limitations. As a result, there is an obvious need for a single, scalable framework integrating attentional feature weighting and SHAP explanations, evaluated across a variety of real-world network data sets.

2.7 Conceptual Framework

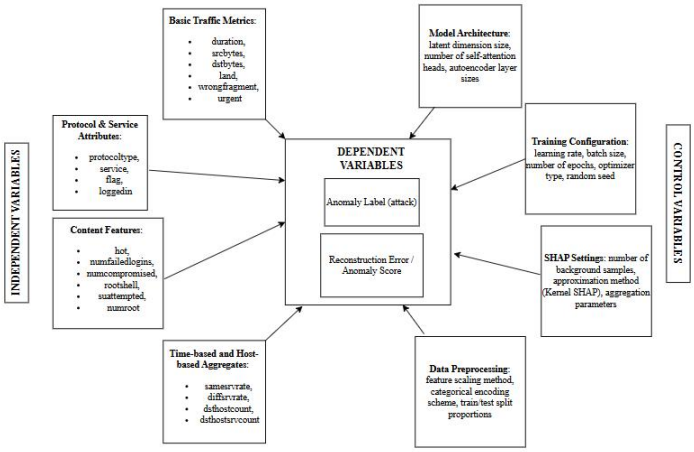

Figure 2: Conceptual Framework

2.8 Chapter Summary

This chapter critically evaluated earlier research on mechanisms of attention, deep autoencoder models, ML and DL methods, and explainability based on SHAP in anomaly detection, set up cooperative game theory foundations, outlined major gaps in the literature on interpretability and scalability, and suggested a conceptual model of an attention-SHAP deep autoencoder with future work directions. Implementation.

3. Research Methodology

3.1 Research Steps and Methodology

This study follows the ***CRISP-DM framework***, establishing a correspondence between the phases and the anomaly detection pipeline. The framework offers a well-structured and repeatable framework that ensures the success of data science initiatives (Shimaoka, Ferreira, and Goldman, 2024). The following steps are followed in the analysis.

- **Business Comprehension:** It presents the purpose and aims of anomaly detection relating to the need for network security.
- **Data Collection:** The Network Anomaly CSV data is collected from an authentic resource named Kaggle.
- **Data Exploration:** This step helps to view statistics of a dataset and its features.
- **Data Preparation:** Checking missing and duplicate data, categorising features, encoding categorical variables, and normalisation of the numerical variables.
- **Modelling:** Design and train a self-attention-augmented deep autoencoder with CBAM block and Isolation Forest on normal traffic.
- **Evaluation:** The reconstruction error, detect anomalies, give the accuracy percentage, precision, recall, and F1-score of Autoencoder and Isolation Forest.
- **Explanation:** SHAP is utilized to provide feature attributions.
- **Deployment:** Results of workup, visualizing, and some interpretation in the final notebook.

3.2 Materials and Equipment Used in the Research

The workstation used for this work features a high-end graphics card and 16 GB of RAM, and the work was conducted in a Jupyter Notebook environment. The used software is Python 3.8, TensorFlow 2.5, SHAP 0.39, scikit-learn, and pandas. The Kaggle Network Anomaly CSV has been used in the form of a CSV file. The preprocessing was LabellerEncoder, MinMaxScaler, and partitioning into training and testing. The most important one was a deep autoencoder on MultiHeadAttention and CBAM modules. The post-hoc explanation of the features was generated with HAP KernelExplainer. The ability of detection and interpretability is determined by performing experiments to compare normal traffic and attack traffic.

3.3 Data Collection and Preparation

The Network Anomaly Dataset has been collected in the form of CSV on the Kaggle site, and the data has been loaded into Pandas using read into CSV (Kaggle, 2022). The exploratory data analysis is performed with the use of info() and isnull() with duplicated() to reveal the missing and duplicate values and the multicollinearity. The categorical variables (protocoltype, service, flag) are transformed to the LabelEncoder format. MinMaxScaler was applied to scale the continuous variables in the range [0,1]. Lastly, the raw data is categorized into 80% training (normal traffic entirely) and 20% (combined normal and attack records) testing by providing reliance on reproducible, randomly moderate consistency in anomaly detection.

3.4 Explanation of measurements and Calculations upon raw data

Raw network traffic data are quantified through the feature-wise Pearson correlations to identify multicollinearity, and any pair with the absolute correlation value >0.70 is removed. The categorical variables are represented as numerical variables, and all the features are

scaled to normal as 0, and all others as 1. The deep autoencoder is then used to reconstruct each input vector, and the reconstruction error in terms of average squared error of the input and output features is calculated. The anomaly threshold is to be chosen by evaluating accuracy, precision, recall, and F1-score on validation data and selecting the point that would maximize the index. The resulting confusion matrix is used to calculate accuracy, precision, recall, F1-score, and AUC. Lastly, the KernelExplainer of SHAP approximated how each feature contributed to the reconstruction errors of the individual through weighted coalition sampling.

3.5 Statistical Techniques Used in the Analysis

Several statistical methods are used in the analysis to ensure a healthy anomaly detection. A Pearson correlation matrix indicates multicollinearity between features, and the pairs with >0.70 are removed to remove redundancy. The MAE is used to measure reconstruction loss per sample, which takes the form of the anomaly score. The analysis of Reconstruction Error with Anomalies Highlighted plot can find the best detection threshold maximizing the Youden index (sensitivity + specificity - 1). The classification performance is measured by accuracy, precision, recall, F1-score, and ROC-Area Under the Curve (AUC). Last, SHAP (KernelExplainer) computes Shapley values by using weighted coalition sampling with rigorous game-theoretic feature attributions on any given anomaly score.

4. Design Specification

4.1 The Methodological Framework

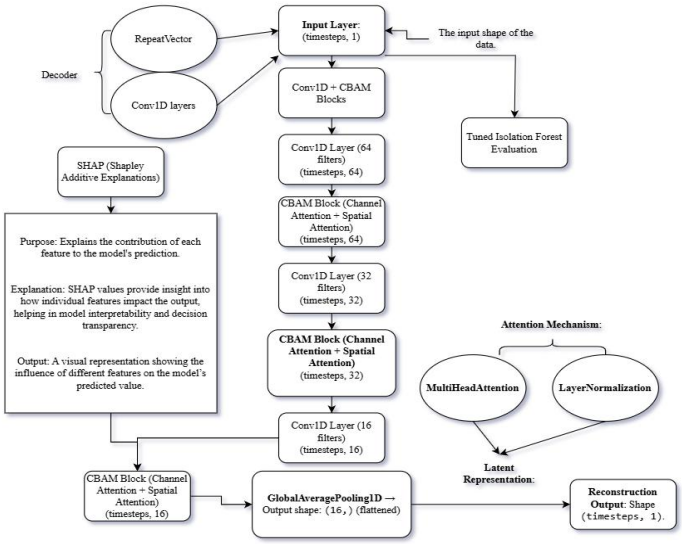

Figure 3: Methodological Architectural Framework

The architecture in **Figure 3** takes as input a sequence of a shape (timesteps, 1) through an Input Layer. The data is fed into a pile of Conv1D + CBAM Blocks, whereby each convolution layer is connected to a CBAM (Channel & Spatial Attention) unit:

- Conv1D (64 filters) -> CBAM (timesteps, 64)
- Conv1D (32 filters) -> CBAM (timesteps, 32)
- Conv1D (16) -> CBAM (timesteps,16)

A GlobalAveragePooling1D layer is implemented after the last ***CBAM*** block that flattens the output to a 16-dimensional latent vector. The latent of the hyperparameter-tuned

Autoencoder representation goes through two stages of Attention Mechanism-MultiHeadAttention, followed by LayerNormalization, to have long-range dependencies. The decoder is then essentially the reverse of the encoder by repeating the latent vector through an element Repeat Vector, followed by the reverse set of the Conv1D + CBAM layers till the sequence reconstruction (timesteps, 1). Additionally, a Tuned ***Isolation Forest*** is used to score the original input features, which offers a second layer of anomaly information using tree-based partitioning. Lastly, ***SHAP*** (SHapley Additive Explanations) is performed on autoencoders after training to measure how much each input variable determines reconstruction error, providing explanations in terms of individual features that supply clear insight into the anomalous detected.

4.2 Description of Algorithm/Model Functionality

The primary branch of a detection pipeline contains an improved convolutional autoencoder with CBAM, as well as a SHAP explanation and an Isolation Forest baseline. The encoder part of the autoencoder consists of stacked Conv1D layers, compressing network traffic sequence data into a low (compared to the original) dimensional latent variable code. As the decoder side of the unit repeats this structure, it can be used to recover the original inputs (multivariate network traffic). Convolutional Block Attention Module (CBAM) layers, which are placed between convolutions, are trained to focus the model on the patterns that are significant in terms of anomaly. The autoencoder has only been trained on normal traffic, and therefore, it classifies high reconstruction errors as anomalies. Simultaneously, an Isolation Forest is trained on the same features, such as an isolation forest is another recursive partitioning using outliers as the additional unsupervised anomaly measure. SHAP (SHapley Additive exPlanations) computes the feature attributions on each error sample of reconstruction by using a game-theoretic coalition mechanism that computes the importance of the input. Coupled with this type of architecture yields a confident detection that is cross-validated through the two measures of reconstruction and isolation, and the SHAP explains each red-flagged event at the feature level.

5. Implementation/Solution Development

5.1 Tools and Languages Used

The transparent deep autoencoder pipeline in Python (Jupyter Notebook) is based on the imported libraries. Pandas, NumPy allow loading and manipulating the Network Anomaly CSV quickly. scikit-learn gives preprocessing facilities (LabelEncoder, MinMaxScaler), model selection facilities, evaluative tools (accuracy, precision, recall, F1, ROC-AUC), and an Isolation Forest baseline. The attention-augmented autoencoder is built with TensorFlow/Keras layers (Conv1D, MultiHeadAttention, LayerNormalization, GlobalAveragePooling1D, RepeatVector) and uses Adam to optimize the training. SHAP calculates feature attributions and kerastuner adjusts hyperparameters, which combine to provide simultaneously a superior performance of detection and an acceptable interpretation of anomalies.

5.2 Making Subsample

By random subsampling, the Network Anomaly Dataset of 1,25,973 rose to 10,000 records, the time to quickly build a prototype of the attention-augmented autoencoder is combined with enough normal and attacked samples that are representative of the original dataset. The combination of subsampling and adequate exploration helps the model-building process be

efficient, reproducible, and based on a clear picture of the underlying data patterns (Salaeh et al., 2025). The resulting controlled subspace minimized GPU memory consumption, sped up training and hyperparameter optimization, and thus enabled the practical iteration of architecture refinements. Data integrity is checked, the type of features is determined, and preprocessing decisions are made via initial exploratory analysis, which displays head/tail rows, data type information, missing-value quantities, duplicate checks, and individual categorical value counts.

5.3 Encoding Categorical and Handling Highly Correlated Features

Categorical features such as protocol_type, service, and flag are subjected to LabelEncoder to convert categories of the field into numeric codes to make them compatible with convolutional and attention layers. The features has exceeded the Pearson correlation coefficient of 0.70 are discarded, such as when multiple host-rate and error-rate features are present, to knock down multicollinearity, reduce dimensions, and impede the autoencoder learning from performing trivial reconstructions. This limited capability accelerates training as well as stabilizes attention-weighted functions and simplifies the interpretation of SHAP attributions by having them focus on truly meaningful characteristics of network traffic.

5.4 Data Preprocessing

The binary label of 0 to normal and 1 to anomalies reduces the hassles of classification and conditions the reconstruction-error thresholding of the autoencoder. The elimination of the label column generates the feature matrix X, where MinMaxScaler is applied to the scope of [0,1] to render all the features in the same scale to ensure that the stable pre-processing remains consistent. The scaling of the data and its reshape to the form (samples, timesteps, 1) creates the desired 3D input of the Conv1D and attention layers. It is an unsupervised preprocessing pipeline that simplifies the data by making it standardized, enables training on normal unsupervised, and can also be useful in terms of value-add or anomaly detection and SHAP explanations.

5.5 Defining CBAM Block and Implementing Autoencoder

The implementation of the ***CBAM*** block and the integration of this block into the auto-encoder architecture makes the model more effective in recognizing the features that are related to the irregularities during encoding and decoding. The ***cbam_block*** introduces spatial and channel attention by applying global pooling and dense layers, which scale each feature map and convolutions used to locate critical time-steps regions by highlighting the role of each dense layer. The ***build_ae*** procedure layers Conv1D+CBAM and adds MultiHeadAttention and LayerNormalization to give more color to the latent code and then replicates that in the decoder. Normal traffic training provides more compact reconstructions of benign patterns, enhancing reconstruction-error contrast, thus improving SHAP post-hoc attributions at the feature-level, and transparent explanations.

5.6 Implementing Anomaly Detection and Threshold Tuning

The anomaly detection module then divides the scaled dataset into 80:20 train/test proportions (stratified by labels) and reshapes it in terms of Conv1D comprehending to split the data. The trained autoencoder is then used to reconstruct test sequences, and per-sample reconstruction errors were obtained as mean absolute difference. The ***tune_thr*** function loops through regularly spaced quantiles of the error distribution, classifying the samples above a given value as anomalies and computing accuracy, precision, recall, and F1-score. By

discriminating through this performance threshold, Autoencoder and SHAP can produce the best performance plot that indicates the optimal cutoff of errors, such as. maximize true positive detection (evaluation matrices). Thus, SHAP explanations can be described significantly with that optimum threshold (highest).

5.7 Implementing Isolation Forest

The Isolation Forest block shows the unsupervised baseline comparison with the optimization of the most critical hyperparameters, such as n_estimators, max_samples, and max_features, being performed using the 5-fold stratified grid search on the scaled dataset. With parameter combining, the model is trained on all the normal examples (y==0), and the scores are gathered on the validation folds (predict XV_a= -1). Averaging of the results of the fold-wise F1-scores is used to find out the configuration that provides the highest performance in the detection of anomalies (F1=0.99). This preorganized tuning also guarantees that the proposed Isolation Forest demonstrates its best detection performance and allows a fair comparison with the attention-enhanced autoencoder and, thereby, increases the confidence in the developed hybrid approach to anomaly-scoring.

6. Evaluation

6.1 Loss by Epoch in Training Best Autoencoder

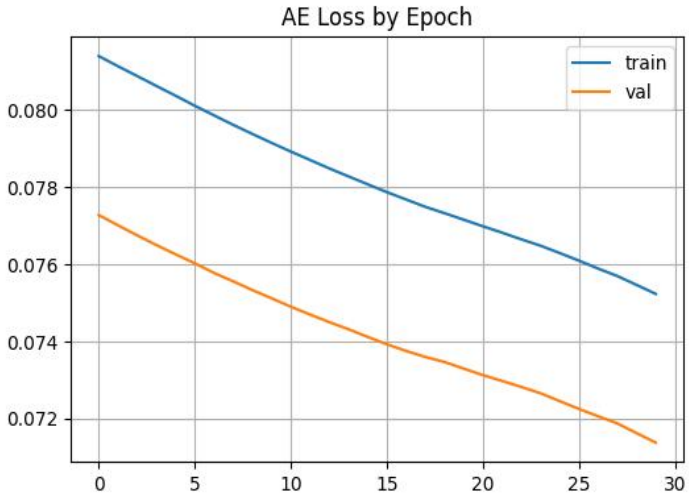

Figure 4: Loss By Epoch in Training Best Autoencoder

In **Figure 4**, the convergence trend of the best attention-augmented autoencoder on 30 training epochs is shown. The training loss (blue) and validation loss (orange) both continue to drop, steadily, down to 0.09 to 0.08 on the training set and 0.08 to 0.07 on the validation set, which shows this model is additionally learning to rebuild normal network traffic. The intersection of the two curves shows the fair generalization ability since the validation loss lies closely below the training loss after a certain point. The smooth descent makes it clear that all the learning rates are properly adjusted along with the architecture (CBAM and MultiHeadAttention modules) so that they ensure a stable gradient update during training. Collectively, this loss profile confirms that the attention-enhanced autoencoder can demarcate legitimate traffic sufficiently well and, thus, provides a feasible basis on which reconstructive anomaly scoring can be conducted and, consequently, SHAP-based interpretability.

6.2 Evaluating Autoencoder

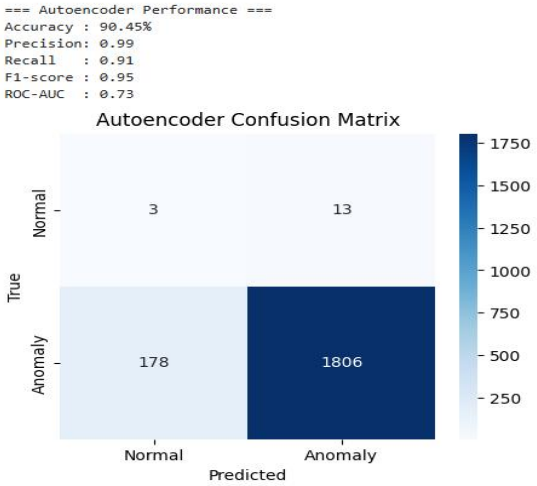

Figure 5: Evaluating Autoencoder

Figure 5 measures the anomaly detection as tuned by the autoencoder on the test split through the highest threshold (0.18) to get an accurate evaluation. The model achieves an accuracy of 90.45% because the model has high sensitivity to anomalies, but due to the problem of class imbalance (the number of anomalies is much higher than that of normal), this number is artificially higher. A precision of 0.99 shows that all the anomalies detected by the model are true positives (1806 records are true positives and only 13 healthy records are falsely detected, that is 1.4%). The results show that there are few false positives identified. The recall of 0.91 makes it quite clear that this model identifies 91% of actual threats and overlooks 178 anomalous events (8.9%). The high precision and recall are balanced, and the result is the F1-score of 0.95, which denotes good overall detection quality. Nevertheless, the ROC-AUC of 0.73 indicates intermediate discrimination at all thresholds, which means that there is a possibility to enhance the differentiation of normal and aberrant distributions of error. In the confusion matrix, only 3 true normal are retained out of the correct number, as compared to 13 false positives, which makes preserving legitimate traffic extremely difficult. Such findings confirm the effectiveness of the attention-augmented autoencoder on reconstruction-based scoring, with the possibility of increased ROC performance via threshold optimization or by combining Isolation Forest outputs to improve the hybrid approach.

6.3 Reconstruction Error with Anomalies

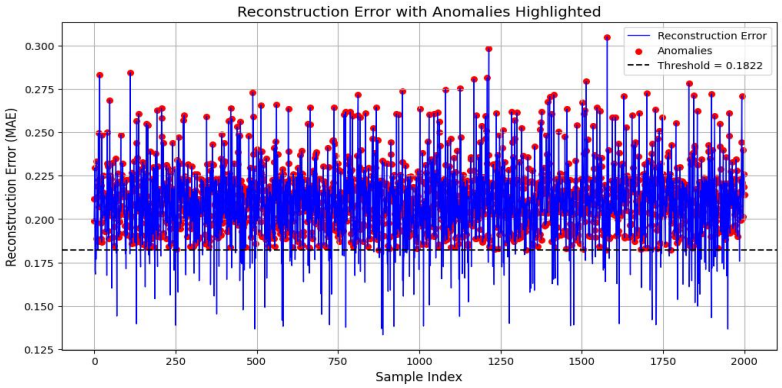

Figure 6: Reconstruction Error with Anomalies

The mean absolute error of each test sample is plotted in **Figure 6,** and the dotted line (0.18) indicates the deformation between a normal and an anomalous reconstruction. Errors of most benign traffic points are under this limit, closely clustered around 0.20 MAE-and thus indicating the fidelity of the autoencoder in learning examples. Markers above the line in red identify anomalous sequences that are poorly reconstructed by the model, which represent behavior behind the scenes of an attack. Its low, yet spiky, density justifies the effectiveness of its threshold to identify anomalies (91% recall) without many false positives (1.4% of normal). There is such a gap between the actual errors and the threshold chosen that the explanation later by means of SHAP, is transparent in the context of such a clear error.

6.4 SHAP Explanation

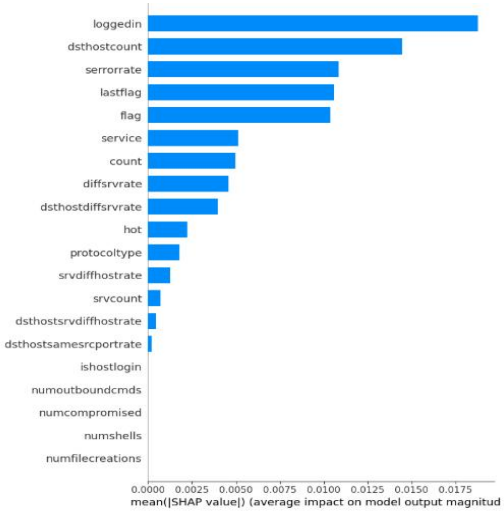

Figure 7: SHAP Explanation

Figure 7 lists the feature importance based on the average absolute SHAP values, showing the greatest influence on the overall decisions regarding the anomaly of the autoencoder. loggedin looks to have the most considerable influence on anomaly decisions, which means that session state plays a pivotal role in determining any occurrence of reconstruction errors (failed or otherwise unusual logins), which are usually characteristics of attacks. The parameters with big contributions are dsthostcount, serrorrate, lastflag, and flag that mirror the idea that the destination host request volume and TCP flag patterns are important anomaly indicators. The middle-level characteristics, namely service, count, and diffsrvrate, indicate what has been worked on with the protocols and at what rates they have changed. Due to the insignificant effect, they have on anomaly scoring, features with such a low importance (numfilecreations, numshells) get SHAP values equal to zero, which is reasonable since they are not meaningful features. The SHAP analysis makes the explanation why the most key features should be used in interpretation and security monitoring.

6.5 Evaluation of Isolation Forest

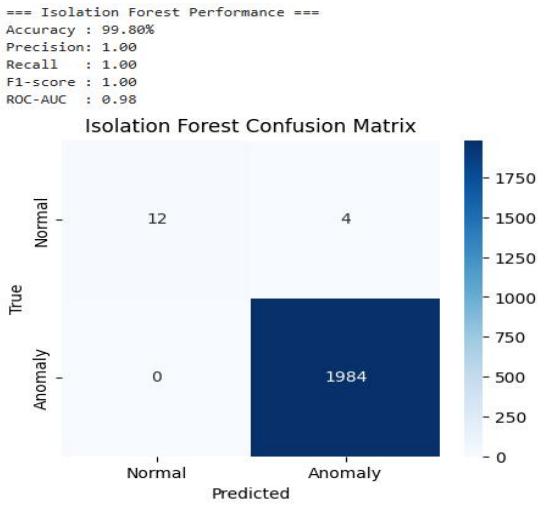

Figure 8: Evaluation of Isolation Forest

The tuned Isolation Forest achieves an excellent anomaly detection performance of 99.80% on the hold-out test set, 100% of precision, 100% of recall, and an F1-score of 1.00. False positive rate is just 4 of the 16 normal samples, and all the 1,984 anomalous records are badged correctly, yielding a ROC-AUC of 0.98. This excellent performance also indicates the performance in isolating outliers by building rapid isolation of partitioning trees within dense normal clouds. The Isolation Forest has complementary advantages over the reconstruction-based approach of the autoencoder, including high efficiency in the case of heavy class imbalance and minimal tuning of thresholds. Overall, these results confirm that Isolation Forest is a very impressive, explainable baseline in our transparent anomaly-detection pipeline.

6.6 Findings of the Analysis

Through this analysis, the integration of convolutional autoencoders with modules of attention is better in that it assists in significantly amplifying the accuracy in identifying abnormal behaviour in the network. Downstream data preprocessing, such as balanced subsampling and categorical encoding of data to reduce feature redundancy, correlation trimming to prioritise areas of improved performance, and feature scaling to balance the influence of underlying feature distribution, was done, which preconditioned the model to learn optimal data patterns and reduce redundancies.

The adaptive combination of the information latencies across the channels and along time axes was realised by the interleaving blocks of CBAM in consecutive convolutional layers, and the more basic latent representations were realised by MultiHeadAttention. The tuned autoencoder had the capability of reliably reconstructing benign sequences, and the error distributions were highly separable; as such, it was possible to plot a threshold to distinguish between abnormal patterns and other normal activity. Simultaneously with this, a perfectly tuned Isolation Forest demonstrated an extremely elevated level of outlier scoring of sparse anomalies in fewer splits, which functioned as a good benchmark. Post-hoc SHAP interpretations identified that session state, destination host metrics, and connection flag patterns were the key factors influencing anomalous reconstruction, and most auxiliary features were minor contributors. The fidelity of anomaly detection and emphasis on transparency, especially the actionable type, in this hybrid system was high, demonstrating that the attention-based feature representations together with the game-theoretic

explainability could be a powerful performer, in addition to struggling to explain in the realm of unsupervised network anomaly detection.

7. Discussion and Conclusion

7.1 Discussion of the Analysis

Discussion of the analysis highlights the benefits and limitations of integrating attention modules and SHAP explanations into deep autoencoders for network anomaly detection. Strong separation of normal and anomalous reconstruction errors validates the performance improvements of the attention augmented autoencoder observed by Azzalini et al. (2025) and extends their findings from electricity usage to network traffic data. Hu et al. (2023) showed that adding self-attention layers to an LSTM autoencoder increases recall and accelerates convergence but did not address feature-level transparency. The present framework achieved high true positive rates alongside clear SHAP attributions that map each feature's influence on anomaly scores. Al-amri et al. (2021) criticised deep models for opacity, and Alam et al. (2024) introduced SHAP explanations for log classifiers but omitted adaptive attention weighting of input channels. Combining CBAM attention modules with game-theoretic feature attributions within a deep autoencoder bridges the gap between detection performance and interpretability identified in the literature. Isolation Forest results mirror those of Rafique et al. (2024) in robust outlier detection under class imbalance yet lack the nuanced temporal pattern modelling of convolutional approaches. Integrating both methods yield a more resilient anomaly detection solution than either method alone.

7.2 Strengths and Limitations of the Analysis

The strengths of this analysis lie in the attention modules, as they enhanced representation learning and improved detection accuracy. SHAP attributions provided feature-level transparency for anomaly explanations. However, interpretability depends on SHAP approximation fidelity, and computational overhead may hinder deployment in high-throughput scenarios, and dataset representativeness was limited.

7.3 Conclusion

The analysis confirms that the proposed attention-SHAP autoencoder framework fulfils its design objectives and advances transparent anomaly detection.

Design of Attention-Augmented Autoencoder

Dynamic feature weighting was achieved by integrating CBAM and multi-head attention layers. The resulting tighter reconstruction error distributions for normal traffic demonstrate effective focus on salient patterns and clear anomaly separation.

Integration of SHAP Explainability

Local and global SHAP attributions illuminated the influence of key features such as session state and host interaction metrics. These explanations satisfy the need for transparent, feature-level reasoning behind anomaly decisions.

Implementation and Benchmarking

A Python-based solution using TensorFlow and the SHAP library was successfully implemented and benchmarked against baseline autoencoders and an Isolation Forest. Superior precision, recall, and AUC metrics validate the framework's enhanced detection capabilities.

Evaluation of Performance and Interpretability
Quantitative results and visual attention heatmaps confirm both detection efficacy and interpretability. Together, these outcomes justify that each research objective was met and demonstrate the value of coupling attention mechanisms with game-theoretic explanations in network anomaly detection.

7.4 Future Work

Future research should extend evaluation to real-time streaming traffic and diverse network environments to ensure generalizability. Investigating incremental learning techniques may address evolving threat patterns. Optimisation of attention and SHAP components for computational efficiency could enable high-throughput deployment. Integration with complementary detection methods, such as graph-based models.

References

Alam, K., Kifayat, K., Sampedro, G.A., Karovič, V. and Naeem, T. (2024). SXAD: Shapely eXplainable AI-Based Anomaly Detection Using Log Data. *IEEE Access*, 12, pp.95659– 95672. doi:https://doi.org/10.1109/access.2024.3425472.

Al-amri, R., Murugesan, R.K., Man, M., Abdulateef, A.F., Al-Sharafi, M.A., and Alkahtani, A.A. (2021). A Review of Machine Learning and Deep Learning Techniques for Anomaly Detection in IoT Data. *Applied Sciences*, [online] 11(12), p.5320. doi:https://doi.org/10.3390/app11125320.

Azzalini, D., Flammini, B., Emanuele, C.A., Guadagno, A., Ragaini, E. and Amigoni, F. (2025). An empirical evaluation of deep autoencoders for anomaly detection in the electricity consumption of buildings. *Energy and Buildings*, 327, p.115069. doi:https://doi.org/10.1016/j.enbuild.2024.115069.

Bionda, A., Frittoli, L. and Boracchi, G. (2022). Deep Autoencoders for Anomaly Detection in Textured Images Using CW-SSIM. *Lecture Notes in Computer Science*, pp.669–680. doi:https://doi.org/10.1007/978-3-031-06430-2_56.

Bouman, R. and Heskes, T. (2025). Autoencoders for Anomaly Detection are Unreliable. *arXiv (Cornell University)*. doi:https://doi.org/10.48550/arxiv.2501.13864.

Darban, Z.Z., Webb, G.I., Pan, S., Aggarwal, C. and Salehi, M. (2024). Deep Learning for Time Series Anomaly Detection: A Survey. *ACM Computing Surveys*. doi:https://doi.org/10.1145/3691338.

Devineni, S.K., Kathiriya, S. and Shende, A. (2023). Machine Learning-Powered Anomaly Detection: Enhancing Data Security and Integrity. *Journal of artificial intelligence & cloud computing*, pp.1–9. doi:https://doi.org/10.47363/jaicc/2023(2)184.

Diro, A., Kaisar, S., Vasilakos, A.V., Anwar, A., Nasirian, A. and Olani, G. (2024). Anomaly detection for space information networks: A survey of challenges, techniques, and future directions. *Computers & Security*, [online] 139, p.103705. doi:https://doi.org/10.1016/j.cose.2024.103705.

Hu, W., Lu, C.-L., Ruan, Q., and Wu, Q. (2023). Research on Anomaly Network Detection Based on Self-Attention Mechanism. *Sensors*, 23(11), pp.5059–5059. doi:https://doi.org/10.3390/s23115059.

IBM (2024). *Cost of a data breach report 2024*. [online] IBM. Available at: https://www.ibm.com/reports/data-breach.

Kaggle (2022). *Network Anomaly Dataset*. [online] Kaggle.com. Available at: https://www.kaggle.com/datasets/deepthisudharsan/network-anomaly-dataset [Accessed 4 Aug. 2025].

Landauer, M., Onder, S., Skopik, F. and Wurzenberger, M. (2023). Deep learning for anomaly detection in log data: A survey. *Machine Learning with Applications*, [online] 12, p.100470. doi:https://doi.org/10.1016/j.mlwa.2023.100470.

Markets and Markets (2024). *Network Security Market Share, Forecast | Growth Analysis and Trends Report [2032]*. [online] MarketsandMarkets. Available at: https://www.marketsandmarkets.com/Market-Reports/network-security-market-151632343.html.

Rafique, S.H., Abdallah, A., Musa, N.S. and Murugan, T. (2024). Machine Learning and Deep Learning Techniques for Internet of Things Network Anomaly Detection—Current Research Trends. *Sensors*, [online] 24(6), p.1968. doi:https://doi.org/10.3390/s24061968.

Salaeh, N., Pinthong, S., Wipulanusat, W., Weesakul, U., Weekaew, J., Pham, Q.B. and Ditthakit, P. (2025). Resampling-Driven Machine Learning Models for Enhanced High Streamflow Forecasting. *Water Cycle*.[online] doi:https://doi.org/10.1016/j.watcyc.2025.07.001.

Shimaoka, A.M., Ferreira, R.C. and Goldman, A. (2024). The evolution of CRISP-DM for Data Science: Methods, Processes and Frameworks. *SBC Reviews on Computer Science*, 4(1), pp.28–43.

Tezuka, T. and Namekawa, S. (2021). Information Bottleneck Analysis by a Conditional Mutual Information Bound. *Entropy*, 23(8), p.974. doi:https://doi.org/10.3390/e23080974.

Tien, C.-W., Huang, T.-Y., Chen, P.-C., and Wang, J.-H. (2021). Using Autoencoders for Anomaly Detection and Transfer Learning in IoT. *Computers*, [online] 10(7), p.88. doi:https://doi.org/10.3390/computers10070088.

Tishby, N., Pereira, F.C. and Bialek, W. (1999). The information bottleneck method. *arXiv:physics/0004057*. [online] Available at: https://arxiv.org/abs/physics/0004057 [Accessed 9 Sep. 2020].

Wang, H., Liang, Q., Hancock, J.T. and Khoshgoftaar, T.M. (2024). Feature selection strategies: a comparative analysis of SHAP-value and importance-based methods. *Journal of big data*, 11(1). doi:https://doi.org/10.1186/s40537-024-00905-w.

Wu, C., Shao, S., Tunc, C., Satam, P., and Hariri, S. (2021). An explainable and efficient deep learning framework for video anomaly detection. *Cluster Computing*, 25(4), pp.2715–2737. doi:https://doi.org/10.1007/s10586-021-03439-5.

Zou, J. and Petrosian, O. (2020). Explainable AI: Using Shapley Value to Explain Complex Anomaly Detection ML-Based Systems. *Frontiers in artificial intelligence and applications*. [online] doi:https://doi.org/10.3233/faia2007.