UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

University of Pretoria

Dr. Mike Nkongolo Wa Nkongolo

Length (Unlimited). Including the cover page, table of contents, list of tables, list of figures, and reference page(s).

Type: Group Assignment (Not more than 4 students per group). Submit 1 PDF file per group. The page must include all students' full names, student numbers, and the group Nickname.

Submission Date: 07 November 2025

Submission Method: Online only

Scenario. You have recently been appointed as a Senior Data Scientist at a leading Cybersecurity firm in Norway, specializing in Advanced Threat Intelligence and Malware Defense (ATI-MD). The company is facing an increasing wave of ransomware attacks that are becoming more sophisticated, evasive, and adaptive. Traditional detection systems have shown limitations, particularly in explaining decision-making processes to security analysts and regulatory bodies.

Mission. To design and implement an Advanced Machine Learning Framework (AMLF) that integrates Large Language Models (LLMs) with Explainable Artificial Intelligence (XAI) to enhance both detection accuracy and interpretability.

Specifically, you are required to:

1. **Model development.** Train and evaluate BERT, DeBERTa, and RoBERTa models on ransomware-related datasets.

2. **Model enhancement.** Improve their performance and interpretability using SHAP (SHapley Additive Explanations) and LIME (Local Interpretable Model-Agnostic Explanations).

3. **Framework integration.** Develop a unified LLM-XAI framework that balances predictive power with explainability, ensuring security analysts can trust and understand the model outputs.

4. **Evaluation and reporting.** Compare models using evaluation metrics while also evaluating the quality of explanations generated by SHAP and LIME.

5. **Practical relevance.** Provide insights into how your framework can be deployed in real-world ransomware defense systems to support proactive threat detection, forensic investigations, and compliance with European cybersecurity regulations (e.g., GDPR, NIS2 Directive).

Learning Objectives

By completing this assignment, you will:

(i) Apply real-world data preparation techniques

(ii) Perform exploratory data analysis and visualization

(iii) Encode and preprocess data for LLM, XAI, and ML modeling

(iv) Train and evaluate the classification model

(v) Generate and interpret model evaluation metrics

Note: Include all plots in your notebook and PDF report with appropriate titles, labels, and brief captions explaining what each plot reveals. In the experiments, compare the performance of the selected algorithms.

Submission Instructions. Each student must submit the following:

Report (**PDF and Word versions**). Your report must include:
o Introduction and problem overview
o Summary of each data preparation step
o Visualisations with captions
o Model choice and justification
o Evaluation metrics and interpretation
o Key findings and recommendations for the Norwegian Cybersecurity Firm (NCF)

Preprocessed Datasets (**CSV versions**)
o Submit a .csv file of your final, preprocessed datasets (after encoding and cleaning, and embeddings).

o File name: **yourname_preprocessed_NCF_data.csv**
Jupyter Notebook or Python Script
o Submit your complete notebook or .py file showing all steps.
o Make sure your code is clean, well-commented, and follows the task sequence.

Suggested Report Structure (Unlimited Number of Pages)
Cover Page (1 page)
Title: e.g., **Ransomware Detection Using LLM and XAI Techniques**
Group Name, Students IDs, Course, Submission Date
Assignment 4

1. Introduction

o Brief Overview of the dataset and the goal (references required)
o Purpose of the assignment
o Outline of the report structure (provide a graph illustrating how the entire assignment is structured)

2. Data Preparation (report data preparation steps)
2.1 Data Cleaning (report cleaning process)
o Description of missing/unknown values if any
o Strategy and justification of handling inconsistencies
o Code summary and cleaned dataset overview
o Feature categorization table (numerical vs. categorical)
2.2 Embeddings and attention weights visualizations
o Description of LIME and SHAPE results
o Sample data before/after embeddings
2.3 Basic Statistics (embedded features correlations)
o Summary of LIME and SHAPE interpretability
o Attention and loss analysis
2.4 Data Visualization
o 4 visualizations (histogram, bar chart, boxplot, heatmap, pie chart, triangular visualization, etc.)
o Interpretation of insights from each datasets using research questions
o For all numerical variables, the data distribution must first be examined to assess skewness. Plot the distribution of variables exhibiting significant skewness, normalized them using appropriate transformation techniques such as logarithmic, square root, or Box-Cox transformations to reduce skewness. Plot the normalized distribution variables reducing skewness. After normalization, all numerical variables must be scaled (e.g., using Min-Max) to bring them into a comparable range. Finally, generate a plot to visualize the distribution of all normalized and scaled variables in a single graph(s), facilitating comparison across features and categories for both datasets. Discuss the impact of this process in real-life for the NCF.

Normalized Features

2.5 Preprocessing and Encoding
o Transformations applied

o Encoding steps and final dataset readiness

3. LLMs (BERT, RoBERTa, and DeBERTa)
3.1 Model Training, Choice, Evaluation, and Comparison
o ML Algorithm selected
o Reason for selection
3.2 Model Evaluation, and comparison with selected LLMs
o Metrics results: Accuracy, Precision, Recall, F1 Score, computational time, and ROC-AUC, **attention weights, and loss**
o ROC Curve plot with brief analysis
3.3 Interpretation
o Discussion on how embeddings influenced results
o Final reflections on model performance
4. Conclusion
o Recap of key findings
o Limitations, and possible improvements

Appendices (**Required**)
o Full code listings (if not inline in report)
o Extra plots or detailed tables

Tips
o Use figures and tables efficiently (combine multiple visuals into one if needed).
o Keep code snippets short in the main report; link or refer to full code in the appendix.
o Use concise bullet points or tables for explanations where appropriate.

Referenced Information. To be able to complete this assignment successfully, the student should reference the sources.

Due Dates. **ASSIGNMENTS SUBMITTED LATE WILL NOT BE MARKED. NO EXCEPTIONS, NO EXCUSES.**

Submission Instructions. This is a GROUP ASSIGNMENT. All assignments will be submitted to Turnitin to assess the originality or the similarity of assignment content. If duplication is found between different sets of project documentation, then the group involved will be penalized.

While AI tools like ChatGPT can assist you, it is crucial that your submissions reflect your own understanding and analysis. Copying or heavily relying on AI-generated content undermines your learning experience and academic integrity. **The use of AI tools as a supplement to your own thinking will be penalized**.

Assignment Layout and Appearance

No handwritten work will be accepted. The manuscript should contain a title page (should be captivating), students names, student numbers, the course name and instructor name. The manuscript should contain an introduction, body, and conclusion. All graphs, images or diagrams should be clearly visible.

o Content on the graphs, images or diagrams should be clearly legible.
o Graphs, images or diagrams should be clearly labeled and entitled.

Assignment Referencing
o References are required for this assignment.
o If a student makes use of any sources, it will be required for students to reference the sources used.
o If a student makes use of direct quotations from any source, then it will be required that a student makes use of Harvard referencing and appropriate citation mechanisms.

Penalties. Penalties will also apply to issues related to spelling, punctuation, grammar, and **figures or data manipulations.**

Problem Statement
Recent advances in NLP highlight the potential of LLMs, such as BERT, RoBERTa, and DeBERTa, to capture complex linguistic and semantic patterns from ransomware artefacts, including logs, metadata, and ransom notes. When integrated with XAI methods like SHAP and LIME, these models provide transparent and interpretable insights into classification decisions, addressing issues of biased detection and model explainability. In this assignment, you will implement a framework using two datasets from Network Traffic (UGRansome) and Process Memory (PM), and apply semantic modelling through LLM to extract behavioural insights of ransomware. You will design a robust, interpretable, and adaptive ransomware detection system capable of handling evolving threats while offering actionable insights to security practitioners.

Aim
The student must implement the proposed hybrid framework for ransomware classification that integrates XAI, and LLMs. The student must fine-tune pre-trained transformer models on two ransomware datasets, improves classification accuracy and enhances generalisation across diverse operational environments. The student must use BERT, RoBERTa, DeBERTa, LIME, and SHAPE in identifying anomalies in network traffic data, analysing ransom notes, and classifying malicious communications. The student must propose interpretable and efficient transformer-based models to tackle emerging ransomware threats.

Task
The student will use two predominantly numerical ransomware datasets to propose a multistage method to adapt LLMs for ransomware classification (Benign Vs. Ransomware). Numerical features from these datasets must be first preprocessed and transformed into textual tokens using discretisation and binning techniques, enabling compatibility with transformers' architecture (see LLM code uploaded on Kaggle under UGRansome to use as a template).

The student will perform numerical embeddings and positional encodings and map these tokens into dense vector spaces, preserving contextual relationships, which are then input to fine-tuned transformer models such as BERT, RoBERTa, or DeBERTa. These models

capture complex dependencies within the given network traffic and process memory datasets.

The models are trained on labeled ransomware features, employing tokenization to represent transformed numerical variables and their contexts as distinct tokens. Model performance is evaluated using loss, attention weights, ROC-AUC metrics, precision, recall, computational time, and F1-score, providing a comprehensive assessment of classification effectiveness.

The student should address class imbalance while ensuring comprehensive ransomware detection. The student should plot the loss function that guides optimization during LLM training, attention weights that enhance interpretability by providing insights into the model's rationale for classification decisions.

The student must apply and plot post-hoc XAI techniques such as Shapley Additive exPlanations (SHAP) and Local Interpretable Model-agnostic Explanations (LIME) to identify key features, whether tokenised inputs or embedding dimensions, that most influence ransomware classification decisions. The student must demonstrate that integrating XAI improves the transparency and trustworthiness of the model's predictions.

Note: The student must demonstrate numerical-to-text transformation, LLM fine-tuning, rigorous evaluation, and explainability of the framework for ransomware classification. The interpretable nature of the proposed model must enhance the classification capabilities of emerging ransomware threats. This framework is evaluated on two given complementary datasets: Network Traffic (UGRansome) and Process Memory (PM). UGRansome captures command-and-control activities external to the system, while PM reveals internal behaviors and traces left by malicious activities within the monitored environment. Together, these datasets will enable robust and comprehensive ransomware classification.

Data Visualization and Communication
Provide visualization supporting the following:

Your report will address the primary question of *how XAI and LLM can be integrated to improve the interpretability of ransomware classification*?

Your report will attempt to answer sub-research questions derived from the primary question stated as follows:

1. What are key features from UGRansome and PM datasets crucial for ransomware classification?
2. How do XAI techniques like LIME and SHAP improve the interpretability of ransomware classification models?
3. How does the proposed framework (LLM-XAI) perform compared to traditional methods (E.g., K-Nearest Neighbour (K-NN), Reinforcement Learning (RL), and Recurrent Neural Networks (RNN))?
4. What are the limitations of applying XAI techniques (LIME Vs. SHAPE) to ransomware classification using LLMs (BERT, RoBERTa, and DeBERTa)?

5. To what extent do specific feature extraction algorithms (E.g., AutoEncoder) enhance the interpretability and precision of ransomware classification?

The objectives of your report are to:
• design a hybrid framework that transforms ransomware related data values into tokens using techniques such as discretization and binning, thereby converting them into suitable input for LLMs,
• apply feature selection to recognise the most relevant features from the network traffic and process memory datasets,
• fine-tune transformer-based LLMs on ransomware-specific datasets to enhance classification performance and generalisation,
• assess and evaluate the hybrid framework performance using metrics such loss, attention weights, F1-score among others,
• incorporate LIME and SHAP to interpret LLMs' predictions and reveal the contribution of the underlying features to the classification decisions,
• compare ML and LLMs performance in classifying data instances into either ransomware or benign,

The following are the key deliverables of this assignment:
1. LLM–XAI Integration. Combining BERT, RoBERTa, and DeBERTa with SHAP and LIME to improve interpretability in ransomware classification.
2. Dual-Source Feature Learning. Using both network traffic (external behaviour) and process memory (internal traces) for comprehensive ransomware detection.
3. Semantic Modeling via LLMs. Capturing contextual semantics in ransomware patterns to provide behavioural insights beyond traditional ML methods.

Experimental Datasets
The Network Traffic dataset (UGRansome) is the first dataset to consider in this assignment. UGRansome, is a recent dataset used for ransomware classification. Besides normal/benign traffic such as UDP Scan, Port Scanning, Scan, and SSH, it includes malicious traffic like Blacklist, Botnet, DoS, NerisBotnet, and Spam. The dataset captures ransomware communication patterns, providing network flow-based features such as bytes transferred, malicious addresses, duration, and protocol flags (Figure 1).

Column	Description	Type
Time	Timestamp of network events	Numeric
Protocol	Network protocol	Categorical
Flag	Connection status flag	Categorical
Family	Ransomware family	Categorical
Clusters	Ransomware cluster ID	Numeric
SeedAddress	Formatted ransomware address	Categorical
ExpAddress	Original ransomware address	Categorical
BTC	Ransomware Bitcoin transaction	Numeric
USD	Ransomware financial loss in USD	Numeric
Netflow Bytes	Bytes transferred in network traffic	Numeric
IP	IP address linked to event	Categorical
Threats	Malware	Categorical
Port	Network port number	Numeric
Prediction	Label class (1 Ransomware, and 0 Benign)	Categorical

Figure 1. UGRansome Features

The Process Memory (PM) dataset is the second dataset to use in evaluating the proposed framework. The dataset captures system-level behaviours of ransomware during execution to enable a deeper understanding of internal ransomware characteristics beyond network-level features. Both the UGRansome and PM datasets are designed for ransomware detection, yet they differ in their feature domains and analytical scope. UGRansome captures network-layer features, emphasising flow-based attributes like byte count, protocol flags, and malicious endpoints to detect and classify ransomware communication patterns (Figure 1). In contrast the PM dataset emphasises host-level behavioural features, such as access privileges, memory usage, and ransomware traces derived from sandboxed execution environments (Figure 2). While UGRansome enables classification based on external network behaviour, the PM dataset provides insights into internal execution behaviour (Figure 2).

Column	Description	Type
r	Read privilege	Numeric
rw	Read-Write privilege	Numeric
rx	Read-Execute privilege	Numeric
rwc	Read-Write-Copy privilege	Numeric
rwx	Read-Write-Execute privilege	Numeric
rwxc	Read-Write-Execute-Copy privilege	Numeric
family	Malware/ Ransomware family	Categorical
label	Class (1 for malware 0 for Benign)	Numeric
SHA256	Record authentication signature	Categorical

Figure 2. PM Features

In this assignment, students are required to integrate natural language-based threat analysis by using pretrained embeddings to represent malware behavior and attack signatures.

These embeddings are fed into classical ML classifiers and LLMs to evaluate detection performance. Additionally, Explainable AI techniques such as SHAP and LIME are applied to interpret the classification outcomes and highlight the key features influencing model decisions. The approach is validated using labeled ransomware datasets, with evaluation metrics.

LLM XAI Model

In the proposed framework, LLMs, including *BERT, RoBERTa, and DeBERTa*, are integral to the feature embedding and classification process. RoBERTa and DeBERTa are both enhancements of BERT, designed to improve performance and address limitations in the original BERT architecture (Figure 3). This model employs input encoding by separately processing numerical and categorical features extracted from process memory dumps and network traffic datasets. For PM, the tokens represent various memory access privileges, denoted by *r, rw, rx, rwx, and rwxc*. These tokens capture different read, write, and execute permissions relevant to the ransomware's operation.

Meanwhile, UGRansome features include identifiers such as Internet Protocol (IP) addresses, and the ransomware's Bitcoin (BTC) transaction amounts extracted from network logs. The input sequence to the transformer model is tokenised. Each token in this sequence is mapped to a learned embedding vector (Figure 3). To differentiate between the modalities, segment embeddings are assigned: *EA* corresponds to tokens derived from process memory, while *EB* is assigned to tokens from network traffic features (Figure 3). This forms a segment embedding sequence such as *EA, EA, . . . , EB, EB*. Positional embeddings *Pi* are then added to maintain the order of tokens within the input sequence, preserving the temporal and structural context of the features. The final input representation fed into the transformer layers is obtained by summing the token embeddings, segment embeddings, and positional embeddings (Figure 3). This combined embedding passes through stacked self-attention layers, enabling the model to learn contextualised representations that capture complex interdependencies across features. Specifically, the output embedding corresponding to the *[CLS]* token encodes the aggregated contextual information of the entire input sequence. This embedding is forwarded to a classification head, which produces logits corresponding to the classes: ***Ransomware and Benign*** (Figure 3).

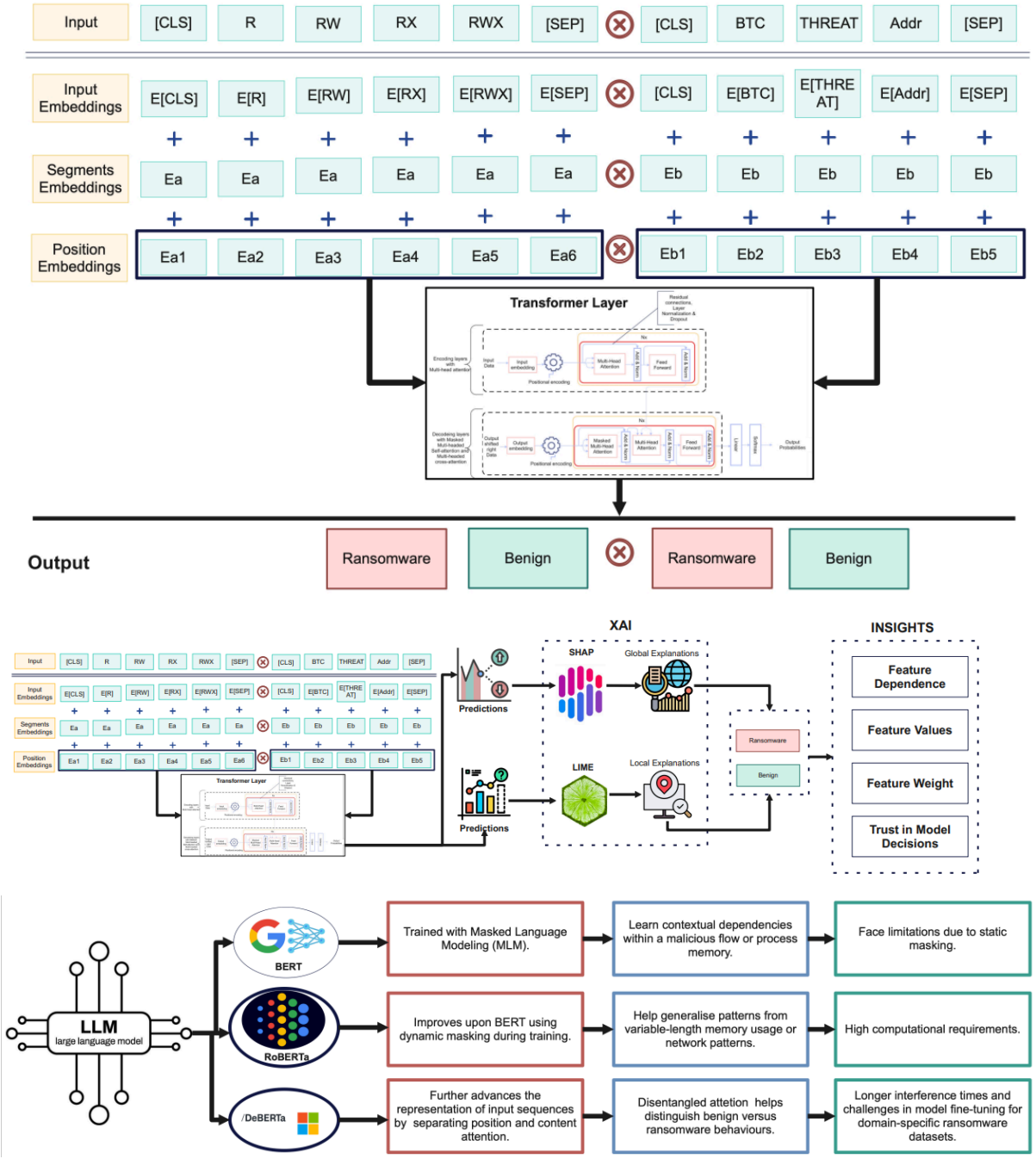

Figure 3. LLM XAI Models

This design facilitates robust classification performance across diverse ransomware samples. Furthermore, the integration of XAI techniques with the transformer model allows the extraction of interpretable embeddings and the derivation of feature importance vectors. These vectors enable the identification and analysis of adversarial ransomware behaviours, improving transparency and trustworthiness in the classification process.

RUBRIC

Category	Marks	Excellent	Good	Fair	Poor
Framework Design	15	Comprehensive problem framing, strong rationale for LLM–XAI integration, well-motivated framework design. (**13–15**)	Adequate framing with some clarity in framework design. (**10–12**)	Limited vague explanation and understanding. (**7–9**)	Minimal or incorrect understanding of the problem. (**0–6**)
Processing & Numerical-to-Text Transformation	10	Innovative discretisation, binning, tokenisation, embeddings, and encoding applied correctly to both datasets. (**9–10**)	Preprocessing applied with minor weaknesses. (**7–8**)	Preprocessing attempted but flawed. (**5–6**)	No or incorrect preprocessing. (**0–4**)
Model Implementation & Fine-Tuning	20	Correct implementation and fine-tuning of BERT, RoBERTa, and DeBERTa with training evidence. (**18–20**)	Implements all three models with *fine-tuning*. (**15–17**)	Implements models with weak tuning. (**10–14**)	Models missing; incorrect implementation. (**0–9**)
XAI Integration (LIME & SHAP)	15	SHAP and LIME are integrated, providing clear interpretability insights with *strong visualizations*. (**13–15**)	SHAP and LIME applied with *partial insights*. (**10–12**)	Method applied incorrectly. (**7–9**)	Missing XAI integration. (**0–6**)

Evaluation	15	Comprehensive evaluation with loss, accuracy, precision, recall, F1, ROC-AUC, computational cost, *attention weights*; compared with ML baselines. (**13–15**)	Evaluates with ROC-AUC. (**7-9**)	Few metrics used. (**4–5**)	No meaningful evaluation. (**0–6**)
Visualizations & Communication	10	High-quality plots (*loss curves*, *attention maps*, *SHAP/LIME importance*). Clear structured results. (**9–10**)	Adequate visualizations, but limited clarity/variety. (**7–8**)	Few or unclear visualizations. (**5–6**)	Irrelevant visualizations. (**0–4**)
Comparative Analysis & Discussion	10	Critical discussion answering research questions. Strong contextual insights. (**9–10**)	Discussion covers most questions but lacks depth. (**7–8**)	Superficial discussion. (**5–6**)	No meaningful analysis. (**0–4**)
Report Quality & Academic Rigor	5	Well-structured, coherent, correct references, academic tone. (**5**)	Minor gaps. (**4**)	Disorganized, weak referencing. (**3**)	Very poor structure, little academic rigor. (**0–2**)