Designing an LLM-Augmented Framework for Security Evaluation and Policy Recommendation

by

Md. Salman Pasha
24141081
KM Abrar Ahsan
24141119
Rumman Nodi
23241082
Shawana Maliha
22101117
Md. Eousuf Siddiquee
24141082

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2025

Declaration

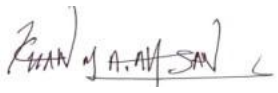
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Md. Salman Pasha

24141081



KM Abrar Ahsan

24141119

Rumman Nodi

23241082

Md. Eousuf Siddiquee

24141082

Shawana Maliha

22101117

Approval

The thesis/project titled "Designing an LLM-Augmented Framework for Security Evaluation and Policy Recommendation" submitted by

1. Md. Salman Pasha (24141081)
2. Shawana Maliha (22101117)
3. Rumman Nodi (23241082)
4. Md. Eousuf Siddiquee (24141082)
5. KM Abrar Ahsan (24141119)

Of Summer, 2025 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 10, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain

Associate Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Associate Professor and Chairperson
Department of Computer Science and Engineering
Brac University

Acknowledgement

To begin with, we would like to convey our utmost gratitude to the Almighty, the ultimate source of wisdom and power who has helped us find our way through this journey. This project would not have been possible without all the help and the support we have received. It is our great pleasure to thank our supervisor, Dr. Muhammad Iqbal Hossain sir who has encouraged us and given us his everlasting support. The warmest of appreciation is for Mr. Moin Mostakim sir for his excellent guidance, encouragement and unwavering assistance that helped us immensely in completing this work.

Abstract

As organizations continue to accumulate more data in the digital platforms, it becomes difficult to safeguard sensitive data. The authorization and authentication factors restrict the use of critical systems, but traditional IAM can't scale with the new breed of cyber-threats like credential theft, phishing and AI attacks. The motive of this paper is in building security systems more resilient and intelligent to deal with any kind of malicious attacks and generating decisions with the application of Large Language Models (LLMs) augmented with advanced AI driven techniques. The project was initially focused on the specified access control factors of Identity and Access Management (IAM) and evaluating policies with providing security scores connected to web interfaces to analyze vulnerable factors beforehand. This was incorporated to a hybrid AI architecture consisting of a small BERT-tiny model optimized to detect security anomalies quickly and larger transformer based models LLMs (Mistral-7B and Gemma3-270M) that can be deployed to explain problems in fine detail and generate remediation strategies that can be executed. Experiments on larger actual datasets showed BERT-tiny achieved a remarkable accuracy of 90.12% for detection and 82.45% for malicious type differentiation. The focus applied on hyperparameter tuning, multi layer approach optimization, and class imbalance adjustments ensured robustness and generalization. Although certain limitations remain, especially in aspects of response latencies and the computational overhead, the present work is a step towards demonstrating the radical potential of an LLM approach in building systems that think faster, explain better, and adapt smarter to emerging digital threats.

Keywords: Artificial Intelligence(AI), Large Language Models (LLMs), Identity and Access Management (IAM), BERT-tiny, Mistral-7B, Gemma3, Security evaluation, Hyperparameter tuning, Policy recommendations, Hybrid architecture

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background

In the present digital world, organizations now heavily rely on digital platforms and interconnected systems in order to regulate their day-to-day operational activities. Different sectors including healthcare, banking, education etc depend on internet infrastructure to transform how they manage and store confidential information along with accessing and sharing data. In the light of emerging cyber threats, it has become a critical challenge to ensure secure access to digital assets for different organizations. Organizations have become susceptible to major risks through digital infrastructure dependency as cybercriminals purposefully take advantage of system vulnerabilities to gain access to confidential information.

Traditional defenses like firewall, IDS and MFA provide baseline security but are not fully capable of dealing with dynamic threats and adaptive attacks [18]. Artificial Intelligence (AI) and Large Language Models (LLMs) stands as the fundamental solution which protects sensitive information because it enables organizations to handle digital identities and control system and data access .The proposed AI-driven framework acts as a comprehensive system that manages secure access and monitors user activities across different platforms. It ensures protection through intelligent rules, adaptive learning processes, and automated policies powered by machine learning and large language models.

1.2 Rational Of the Study or Motivation

With the advancement of technology, the cyberattacks have also evolved to different surfaces. The attack surface is growing on a continuous basis, furthering the meaning of the Internet of Things (IoT) as it is becoming more difficult to check on all network devices protecting the privacy securely [11]. On top of that, the adoption of cloud computing has driven the security processes even further complicated as organizations have to handle the identity and access control in the scattered cloud environments to comply with stringent security practices and regulations. As a result, the conventional IAM systems are proving to be less efficient against these much more complex cybersecurity threats. Attackers currently exploit enhanced methods like credential cracking along with social engineering phishing to bypass conventional defensive measures. credential theft where the passwords and usernames of

users are stolen continues to be an extremely popular attack method experienced by organizations. Attacker access to compromised credentials which grant them illegitimate entry into important systems so they can extract data and create disruptions or use stolen credentials to pretend to be valid users. The SolarWinds supply chain attack of 2020 remains one of the most widespread, destructive and complex cyberattacks in history that affected the U.S. government and corporations through massive breach of confidential data [10].

In order to overcome these growing obstacles, access control systems need to be improved in order to accommodate real-time threats and minimize risks in advance. Here, Artificial Intelligence (AI) steps forward to be a game-changing solution. This is because incorporating AI into IAM in addition to the basic IAM functionality results in an increase in ability to forecast, identify, and prevent security threats. Machine learning based AI driven IAM systems use machine learning algorithms to analyze, automate and learn from patterns, detect anomalies and respond to threats 'proactively' ahead of time. These systems can monitor unusual login behavior, identify suspicious attempts to access them through real-time access controls then automatically block access to the affected accounts. Indicatively, Microsoft Active Directory, which is one of the popular IAM tools, included AI to enhance prevention of credential theft and unauthorized access [1]. AI is capable of identifying suspicious patterns of login and preventing brute force or credential stuffing attacks before they can give a user access to the system. Correspondingly, AI can operate under frameworks such as Structured Threat Information eXpression (STIX) to classify and foresee potential attacks to enable organizations to remain ahead of the attackers [33].

Integrating the policy tools with LLMs can essentially provide automated decisions relating the way to handle access and identify verification with learning the contextual information. The patterns observed by the large volumes of text input, such as logs, communications, and documents are learned by LLMs with earned knowledge which lacked in the traditional rule based approaches. Compliance risks are also identified by LLM monitoring user nature of access and can generate a report based on the results for further scope of improvements [32].However, It also brings with it some cautionary aspects that should not be neglected. The reason is, AI can also be used in creating more cyber threats with generation of more information regarding attacking sources. Ensuring accuracy of various ranges of data provided to the system is also necessary to prevent from bias ness. The best types of LLMs are incorporated with AI, but organizations need to maintain a balance between strong security and trust of users by showing what it is doing and closely tracking its application allowing an additional layer of protection against security breaches.

In short, using AI in security systems can bring a major change in how organizations handle and manage their security problems. Augmentation of AI with IAM can help reducing the chance of sensitive information leaking by enforcing smarter, more precise access controls. Large language models translate findings into understandable explanations and recommend likely solutions. With continuous learning, the system improves over time and can scale to support larger, more complex environments.

1.3 Problem Statement

Nowadays, as everything is growing and becoming more digital,, the dependency towards large scale data systems have increased a lot. Different kinds of cyber threats are allowing themselves to bend in with a newer form in infiltrating these systems signalling higher risks. Unfortunately, the traditional IAM policies fail to properly handle these evolving security threats. These security threats include issues encountered because of ensuring real time scalability, complex design structure, user security threat management etc. This is due to the absence of compatibility and integration between old systems and developed AI solutions. As a result, this leads to the inefficiency of organizational operations since the standard threat detection policies used in IAM lack sufficient defense against modern cyber attacks. Moreover, the lack of intelligence that is powered by Large Language Models (LLMs) in the traditional IAM systems inhibits the system to analyze contexts, identify anomalies, and make autonomous security choices. Consequently, security professionals have to encounter a high number of false alerts daily that need immediate attention and allocation of resources. This creates increased risks of data breaches and puts unbearable pressure on security professionals since they have to redirect their resources to deal with potential threats rather than focusing on actual security threats. Due to slight lack of attention to detail, organizations may have to deal with financial and reputational losses ultimately ending with severe legal repercussions. These identified issues can be addressed by the implementation of AI-enhanced IAM model augmented with the help of the LLM as the system requires time-sensitive flexibility and scalability of architecture along with efficient security measures to provide full data privacy and protection. It is seen that the proposed system is resistant as it enhances future security and automated responses to tackle the existing cyber attacks with the aim of protecting vital information.

1.4 Objective

The combination of IAM with LLM augmented systems and AI technology addresses the key functional challenges of the basic IAM systems in terms of large user management and slow false alarms detection as well as adaptability issues. Using AI tools, transformation can be done through automated access control decisions which would lessen the human workload improvising protection. With organizations managing a high rate of data that are related to user activity and authorization patterns, the necessity of combining AI and LLM based techniques can save real world security issues and make decisions based on it.Such innovations play a wide role for the enterprises giving suitable insights and in fewer time. This research will focus on:

- Proposing and building a hybrid framework that integrates Identity Access Management (IAM) practices with LLM-enhanced solutions based on contextual analysis of security threats.
- Reactive Anomaly Detection techniques that can serve towards security enrichment through minimizing unnecessary detection alarms by means of LLM based applications.

- Training classification models on malicious and non-malicious sets of samples to identify attack vectors like XSS, SQL Injection or Command Injection etc. to grasp a proper idea to habituate on further impacts..
- Applying Maintaining Explainable AI (XAI) and LLM interpretability techniques to bring understandable insights building trust among users and professionals.
- Evaluating the system in real world situations across various industries to analyze the applicability and reliability in practice in case of different industrial environments.

1.5 Methodology in Brief

The methodology of this research as shown in figure 1.1 focuses on building an intelligent and explainable cybersecurity framework that integrates machine learning and large language models to enhance identity and access management towards an automated policy-based security evaluation. Since the traditional IAM struggled to deal with newer real time threats, the proposed framework applied both organized data analysis and situational thinking of security configurations to deal with such. Many machine learning models were employed to get a leveled description of security risks with a score with interpreting threat conditions using transformer based models. The dual approach guarantees in giving a suitable result adding reasoning based on it which brings about user satisfaction in cybersecurity environments.

A significant aspect of this methodology lies in the use of smaller, efficient models such as BERT-tiny for rapid anomaly detection with later in-depth analysis focused through the help of larger models like Mistral-7B and Gemma3-270M. The models are fine-tuned through multiple experimental runs to optimize hyperparameters, ensuring stable performance with minimal resource use. The AI models are embedded within a web-based interface that ensures smooth and intuitive interaction for users. Through this tool, users can easily submit their data and explore meaningful visualizations of security insights. The layered and modular structure not only enhances detection accuracy but also promotes adaptability, transparency, and scalability. Altogether, it forms a balanced system capable of assessing risks, recognizing evolving patterns, and generating smarter, adaptive security responses to new and emerging cyber threats.

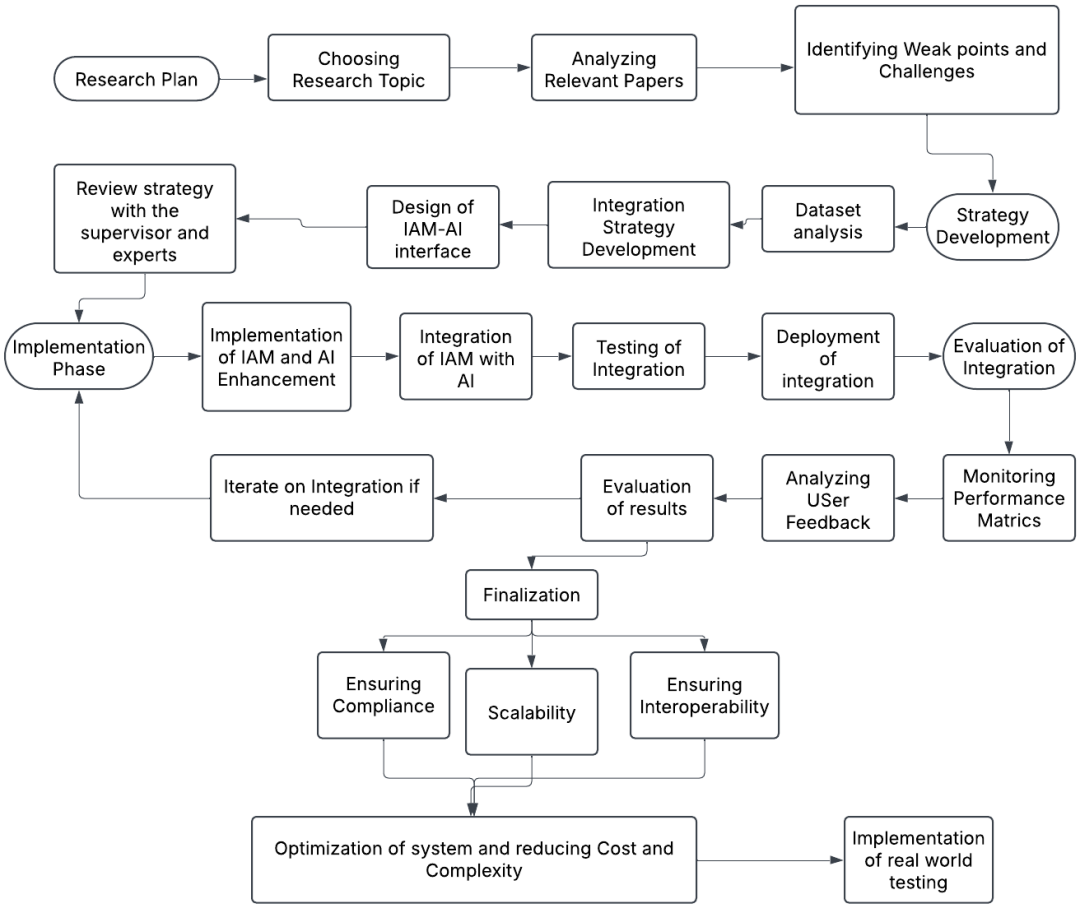


Figure 1.1: Workflow Acitvity Diagram

1.6 Scopes and Challenges

Scopes

1. Security Improvement: The driven systems through LLM can go a long way in enhancing cybersecurity such as detection of unusual behaviours, preventing unauthorized users and establishing authentication processes by conducting advanced language and pattern analysis.

2. Automated Access Control: To grant or deny permissions without needing human interactions, LLMs can analyze data like user intent and adjacent behaviour.

3. Threat detection and response: The AI integrations are aimed at monitoring suspicious activity and detecting abnormalities and acting accordingly or eliminating them even better.

4. Enhanced User Experience: One can ensure an impressive security imposed with the help of AI and make the procedure of authentication easier like adaptive authentication.

5. Regulatory & Compliance Adherence: LLM serves companies to adhere to the compliance regulations through automation of the audit list and the risk control.

6. Cost Efficiency: AI based LLM solutions have the potential to assist in optimization of operational costs through reducing manual operations in identity verification and fraud detection.

7. Interoperability with Legacy Systems: The study can review the systems to ensure that the AI will be able to support the standardized LLM splicing, without interrupting or shutting down critical operations.

8. Deployment in Real Environments: The implementation and testing of the Real-life of the LLM-based security models is bound to test the performance, scalability, and capability of the models in detecting and hypothesizing risks and responding to them.

Challenges

1. The pools of data that LLMs need to be trained properly are large, quality and diversified; this data can be quite challenging to find in security fields.

2. When-ever AI is used to provide access control systems, but the training is based on specific data, which only defines the partial faults of the system, it is discriminatory.

3. The AI system applied to the implementation of methods is vulnerable to attacks, which may adversely affect the level of security of IAM systems.

4. AI is complex since it cannot be integrated in the existing mainframe.

5. Ensuring LLM decision-making helps the administrators to guide system operations and arbitrary problems of the system are resolved in a more productive way without neglecting operational correct accountability.

6. The implementation imposes organizational compliance with security regulations across the globe especially GDPR and HIPAA and NIST by undertaking rigorous legal scrutiny processes.

7. Organizations may hesitate to adopt LLM-driven systems unless sufficient testing, validation, and user trust are established.

8. The installation of the LLM based systems is usually associated with high tech systems and specialists who are highly trained and incurring high costs of implementation to the organizations. The development of practice systems introduces challenges in terms of execution since in the real world, there is a requirement of secure environments that are uncompromising security feeding holes.

Chapter 2

Literature Review

2.1 Preliminaries

In the early stages of cybersecurity, traditional identity and access management (IAM) systems relied heavily on password-based authentication and role-based control mechanisms. In the 1990s, the statical one way method could not anymore live up to expectation since gradually the cyberattackers attained control over the access to the sensitive information easily through different means like brute-force attacks and leaking of passwords through multiple guesses. The authentication process became necessary to develop to mitigate phishing and leakages of information. Examining this, many revolutionary frameworks were utilized by the late 1990s and 2000s. One of the starting was with Role-Based Access Control (RBAC) where a

certain role was allocated to work upon that was required to input to the system to get access. This could not serve the purpose of solving the previous issues of cyberattacks either. The inadequacy later on introduced the then generation with better security solutions working with multi-factor authentication (MFA) and the token-based authentication in the early 2000s which came useful for checking the assurance of the user's identity using more than one means of approval terms. With the digital emergence of cloud computing structured for distributed resources, the pattern of biometrics and extra layer of protection policies itself struggled to keep up with the infrastructure changes and yet started coming to the control of hackers. This resulted in a cloud-native solution for IAM to accommodate the dynamic environments following which brought a new perspective to computer vision with the breakthrough of Artificial Intelligence (AI) and Machine Learning (ML) around the 2010s.

The effect of AI has gained a massive global attention since then. Gradually, AI controlled IAM technologies took over previous rule-based workstyle in handling all credentials using behavioural based control and detecting anomaly in the initial phase. Concepts such as Risk-Based Authentication (RBA) and Zero Trust Architecture (ZTA) emerged, reinforcing the idea that trust should never be assumed but continuously verified. Around the late 2010s and early 2020s, the introduction of Large Language Models (LLMs) such as BERT (2018), GPT-3 (2020), and later Mistral (2023) redefined the landscape of AI in cybersecurity. These models also added some sense of context and generative reasoning, which enabled security sys-

tems to not only identify a threat but also elaborate and suggest suitable mitigation measures. Unlike earlier models that focused solely on pattern detection, LLMs could interpret the semantics of malicious inputs, recognize intent, and generate policy-aligned responses in natural language.

The advancements in the technology area can be only understood with the help of data accustomation provided by the AI-perpetuated implementation. Following that understanding, effort funneled into the creation of a multi-phase AI system with the help of which the security incidents are interpreted, classified, and mitigated with the use of LLMs. An underweight BERT-tiny model performs initial malicious inputs detection, whereas higher-quality LLMs, including Mistral-7B and Gemma, are engaged in the contextual model analysis, providing human friendly explanations, and recommending malicious mitigation. This evolution has seen AI play a different role in cybersecurity changing its task of realizing anomalous detections to policy-noticed reasoning and automated reactions to threats. In addition to identification of potential threats based on LLM, attack patterns may be translated into practical lessons to be applied by administrators who need to make decisions that cut across the technical detection and strategic decision-making gap.

2.2 Review of existing research

In a research, Bolton, Sheikhfathollahi, Parkinson, Basher and Parkinson (2025) explores how large language models (LLMs) can be organized into teams with specific agents working in collaboration to solve problems more effectively than a single working model [23]. The unified work between multiple LLMs can improve reasoning, accuracy and efficiency in complex scenarios making it simpler. Experiments were done with establishing interaction between multiple agents through communication strategies like direct messaging, centralized coordination and role based specialization. Each task was assigned a specific function allowing it to divide into smaller sub tasks and cross checking each other's outputs. It was found that multi-agent systems achieved higher task completion rates with better reasoning consistency compared to single LLMs. The same systems were also capable of correcting the other one and validating its work. The study discovered, however, that it had a number of limitations such as overhead on communication, frequent confusion to message and increased cost in calculation since multiple agents are alive. The other difficulty was to control the context window that led to the loss of information given to the interaction for a long time. In spite of these, the multi-agent collaboration was clearly demonstrated to have high potentials in enhancing the performance of the LLC in multiple domains. The future models should not be focused on only making larger, rather in designing frameworks where sharing of roles with task division and improving coordination protocols and integrating human feedback into agent collaboration. Overall, it can be concluded that multi-agent systems represent a promising step toward building more reliable and scalable AI frameworks which can be capable of handling difficult real-world problems.

A current research of Henke (2025) investigates how a system named AutoPentest can be used to create a more autonomous and efficient penetration testing through exploring large language models (LLMs) [25]. The main concept is to determine if

an LLM agent architecture, based on GPT-4o and LangChain, can assist in adding intelligence and intuition to black-box penetration testing, in which testers know only the IP address of the target. AutoPentest is multi-agent based with the roles of a Planner, a Supervisor, and a number of specialized workers, which are specialized in tasks such as enumeration, injection, privilege escalation, and authentication testing. To enhance the quality of the decision-making, the system combines Retrieval-Augmented Generation (RAG) with a vector database to enable the agents to retrieve domain-specialized knowledge, and also uses tools such as nmap, NVD API lookups, Python scripting, browser automation, and shell execution. The researchers ran AutoPentest on three Hack The Box (HTB) machines (Devvortex, Broker, Codify), which were posted after GPT-4o had a training cutoff to prevent the possibility of contamination with existing solutions. It was found that AutoPentest was able to solve 1526 percent of the necessary subtasks, which was a little better than the manual application of ChatGPT-4o. Nevertheless, the enumeration and vulnerability identification performed well, whereas successful exploitation was frequently a problem of the system, and it is one of the reasons why autonomous approaches are currently limited. The cost analysis showed that the number of themes that AutoPentest was used in amounted to $96.20, which is approximately 9.62 per experiment on average, as opposed to the prior, fixed subscription of ChatGPT Plus, which costs 20. AutoPentest was also more costly, but more scaled because of the increased query limits. The results indicate that pentesting based on LLM can be done and that it is in its initial phases: autonomy was strong with the main difference occurring in the approval of commands to be used in the name of safety but the long-term execution aspect due to repetition and misjudging shell states constrained success. Relative to other projects of the time such as PentestGPT or AutoAttacker, AutoPentest exhibits greater autonomy, well-considered handling of post-cutoff systems, and incorporates RAG, which is why it is a welcome change. Nevertheless, there is still the problem of dealing with interactive tools (e.g. Metasploit), enhancing the success of exploitation, and the ethical issues of dual use. Altogether, AutoPentest demonstrates that the application of LLM agents can help shift penetration testing to scalable, semi-autonomous security testing, which can be useful in future frameworks caring to integrate detection of vulnerabilities with extended security testing and policy advice.

Over the past few years, researchers have been aiming to build an AI powered system that can automate the very time consuming, costly and manual labour heavy process of penetration testing. A promising development named VulnBot was introduced by Kong et al. (2025) exploring multi-agent architecture of LLMs [27]. Here in this system, penetration testing is performed by replicating its manual processes where different tasks are executed by different people. This is where the Multi-Agent architecture comes in handy giving each agent a separate and clear task to execute like assigning an agent to gather the information about the target, another agent for scanning (seeking weak points), and another for exploitation (attempts to break in) etc. These agents organize their activity with the help of a Penetration Task Graph (PTG) that serves as a map to ensure the tasks are accomplished in the correct sequence and nothing valuable is lost on the way. To maintain smooth communication, a Summarizer module can assist the agents in transmitting only the most pertinent information and prevents confusion and the so-called context loss

issue which severely limits the performance of the LLM. The difference with VulnBot is that it does not only assist a human tester but it can run penetration tests on its own. Experimentally, VulnBot outperformed the best models including GPT-4o and Llama3, doing more tasks and even end-to-end vulnerable machine penetration tests with retrieval-augmented generation (RAG). Simultaneously, the authors recognize certain weaknesses: the system continues to have difficulties with complex tasks of exploitation, is not able to utilize images or graphical products of tools, and is only tested in a controlled setting. Nevertheless, VulnBot demonstrates that LLMs are capable of being more than simple question-answer and performing a structured multi-step security task autonomously. In this thesis, VulnBot can serve as an example of how to create structures that integrate LLMs with systematic planning and agent cooperation. As much as VulnBot is focused on finding out vulnerability and exploiting it, the same can be applied to the aspect of security testing and policy suggestions, where various agents can be specialized in risk analysis, mapping of the risks to compliance criteria, and recommending improvements that can be done to it.

Recent work by Wu et al. (2024) investigates the growing concerns in security policies based on large language model (LLM) systems [20]. Rather than only focusing on individual models, the analysis was done with composition of several interacting elements like plugins, sandboxes, web tools, and front-end interfaces. Here, the main goal was to understand how vulnerabilities can arise not just within the model itself but also from the way these different parts communicate and share information. An information-flow based framework was introduced to visualize how data moves surrounding these components. They identify that an LLM system's security depends on two main factors: how the model processes data and how the model exchanges data with other components. To test their approach, they applied it to the OpenAI GPT-4 system and found out several security weaknesses even in its advanced safety design. For instance, GPT-4 could sometimes create external links that led to inappropriate or harmful content leaking private user data through indirect prompt manipulation attacks. These issues occurred though there were safety filters, showing that current protections can still be bypassed through complex interactions between system layers. The researchers showed a real-world example where an attacker could access a user's chat history without directly hacking the model. The study concludes that LLM security testing should go beyond just checking the model itself and focus on the entire system, since weaknesses often come from how different parts work together. This research is important for understanding LLM security as a whole and supports the goal of creating safer, policy-aware AI frameworks.

Vitla (2024) discusses in his publication that AI enhances IAM systems by including anomaly detection capability with adaptive learning characteristics and scalability potential along with regulatory compliance features [19]. Current organizations struggle to execute effective security solutions since traditional IAM systems operate by combining manual provisioning methods with written regulatory codes. The research shows that deep learning operates in real-time protection of system-level attacks by utilizing adaptive authentication methods with behavioral biometrics. Predictive analytics systems must acquire user access requirements for upcoming

periods ahead of compliance audits that review automated access provision generated from affinity-based policies. A dual approach of encryption policies that secure user data links with monitoring systems for dynamic policy enforcement enables both customer trust and regulatory compliance. The AI system performs automated provisioning work that covers 75% of tasks while reducing false positive detections by 40% through its advanced design. Its flexible security framework can adapt to user behavior in real time, improving both system performance and user experience. Automated testing also needs very little user input and can quickly fix problems found in earlier systems. In this research, security and scalability are built into every part of the system, allowing it to make better real-time decisions through continuous learning. The framework also supports privacy-preserving methods while staying scalable, making the overall process more efficient.

In another study, Younas et al. (2024) recommended a highly accurate AI-based model for detecting early Cross-Site Scripting (XSS) attacks[21]. Their approach used advanced feature engineering and machine learning techniques. Based on a Kaggle XSS dataset containing both malicious and benign scripts, they introduced a new feature fusion technique, named LSTF, as a combination of temporal features extracted by a Long Short-Time Memory (LSTM) network with statistical features by Term Frequency-Inverse Document Frequency (TF-IDF). Different machine learning models like random forest, logistic regression etc. were tested and evaluated using the feature set. Additionally, techniques like hyperparameter tuning, Explainable AI (XAI) like SHAP and cross validation were used to understand the model even more. The results indicate that the LSTF feature fusion approach was much better in boosting the detection performance in comparison to the Bag of Words (BoW), TF-IDF, or LSTM features. It is very important to note that the model with the best performance was the Random Forest model with the accuracy of 99%, which is superior to all other models and state-of-the-art approaches that were mentioned in the previous literature. Also, the model was computationally efficient relative to deep learning models, which made it applicable in real-time. The weakness of this study is that it uses a specific dataset on Kaggle, and thus, it needs to be tested on a wider, more realistic and current web traffic data, that the model is being used as a standalone solution, and the model needs to be tested on more advanced, obfuscated XSS payloads, which the model is highly accurate against.

The correlation of how AI and LLM can build a worthy interpretation of the security measures of IAM is presented in a comparative manner by Mustafa et al.(2025) in his article[29]. As the author says, the concept of AI and LLM can have the largest improvement to this point of weakness since the traditional based security takes are no longer responsive enough to the changing cyber threats as the author explains. The literature on the use of LLMs in threats detection, automated incident response, vulnerability assessment and policy generation is also discussed in the study. Nevertheless, the author indicates some problems such as data bias, model interpretability and adversarial attacks during the process of conducting the research. The most problematic issues include prejudice in information and interpretation limitation, and inability to resist attacks, the writer argues, as this decisiveness is vital to reliable clarifiable AI solutions. This research serves as a base of our investigation when it states that the new need of an area of modern security problems requires

the creation of LLC systems, which will be capable of offering a contextual based approach to solutions towards the dynamic by nature of the security issues that are occurring. It is also going to secure transparency in terms of building smarter security management to benefit everyone as well as securing the level of protection. This paper complements the intent of this purpose, which is LLM-augmentation, i.e., explainable AI systems to transparent and adaptable cybersecurity solutions in the IAM systems.

Jaffal, Alkhanafseh, and Mohaisen (2025) conducted a broad survey of increasing the role of Large Language Models (LLMs) in cybersecurity, their practical implementation, weaknesses, and defense strategies [26]. The authors add that the high contextual reasoning and language understanding of the LLM is revolutionizing major fields of cybersecurity including network defense, internet of things, cloud security, and access control management. The work reviewed 223 publications on 2021-2025 dealing with the application of Large Language Models (LLMs) to cybersecurity through PRISMA 2020. It discovered that LLMs are aiding in a variety of issues, including finding intrusion, identifying systems vulnerabilities, identifying suspicious behavior, and automatic responses to the threat. It was also revealed during the study that LLM results in stricter, quicker and more contextually intelligent security system. Nevertheless, they have certain issues, including falling into the traps of attackers, their data being corrupted, and being deceived with malicious suggestions. Simply put, the research indicates that LLMs hold enormous potential with regard to enhancing cybersecurity, but it is necessary to use them with caution and responsibility. To deter them, the authors mention such options as constant red teaming, safety-driven fine-tuning, and the content filtering. This study is closely connected with the aims of the current thesis since it proves that Identity and Access Management (IAM) systems may be enhanced in providing an applicable and operating manner through enhancing their essence with the assistance of LLM. It is in line with the idea that once the LLMs have become a part of IAM, one will be capable of enforcing policy in a more intelligent manner and spot anomalies in real-time and hence help to make an enterprise environment more secure and context-sensitive.

In order to fully explore the intrusion prevention techniques, Polinati (2025) suggests a detailed framework, integrating deep learning and explainable AI, to enhance the intrusion detection system (IDS) [30]. According to the authors, traditional signature-based IDS tend to be incapable of identifying emerging threats, such as zero-day vulnerabilities, as well as Advance Persistent Differentiation (APTs) and adaptive and clever detection engines. Bearing the restrictions in mind, a hybrid AI platform was facilitated and comprising of Autoencoders, Convolutional Neural Networks (CNNs), Long Short-term Memory (LSTM) networks, and Graph Neural Networks (GNNs) was used to have spatial, temporal, and relation pattern in network traffic with ease. The framework was trained and tested using common benchmark data such as KDD Cup 99 and UNSW-NB15, with a significantly greater detection accuracy with lower false positives than other intrusion detection systems used traditionally. Also employed in making decisions by the study, are Explainable AI (XAI) tools that include SHAP and LIME to aid in simplifying the model and instilling confidence in its auto-reasoning. The degree must be particularly perti-

nent to critical industries of finance, healthcare, and government where they need real-time performance, scalability, and transparency most because of the critical nature. The present paper is quite consistent with the existing thesis as it will provide technical justification to the fact that AI-driven anomaly transfer can be successfully implemented in the framework of security systems, and extracting the example of how hybrid deep learning and XAI techniques can automate the policy enforcement notions, enhance the interpretability, and allow making the decisions based on contexts in LLM-enriched IAM-based environments.

AbdelAziz (2024) presents a practical application of AI-based system that refers to a series of vulnerable scanners and combines them with large language models (LLM) to optimize the vulnerabilities that have been identified and those that have been fixed by the cybersecurity system [13]. The author observes that the traditional tools such as Nessus, OpenVAS and Nmap are commonly applied without combining thus giving fragmented results and incomplete security reports. To solve the described problem, the proposed system will combine the output of the scanners and input the LLMs, which will be trained on the cybersecurity principles, including MITRE ATT&CK and NIST. This combination allows the model to better put vulnerabilities into perspective meaning that it is more specific on prioritizing vulnerabilities. The paper also illustrates how the LLM can autonomously narrate weak points to be exhibited, provide solutions, and produce the predictability of compromising by looking at the logs and settings of the system. This practice was experimented using vulnerable simulated systems and it was found out that it enhanced the detection accuracy by around 50 percent and the time spent remediating by 60 percent as compared to the traditional manual reviews. No matter the combined enterprise of AI logic and technical vulnerability detection, it is noted that the process of the whole survey will be more intelligent, adaptive, and effective (Arora, 2017). It is directly related to the concept of creating an Identity and Access Management (IAM) security architecture with the LLM improvement that would be capable of evaluating settings, determining security ratings and promoting policies. The article validates the fact that the incorporation of AI and LLLM has the potential to alter how organizations detect and respond to risks since it incorporates information on different tools and bolsters decision-making. Similarly, the offered IAM framework uses these concepts to better control access control and security policies to create a system that is automated, transparent, and can learn based on the real-time data to enhance the security of the enterprises.

Another breakthrough in the field of software vulnerability detection was acheived by Jean Haurogné, Nihala Basheer and Shareeful Islam (2024) where the authors introduce a new method which consists of a fine-tuned BERT-based Large Language Model (LLM) paired with a detailed transparency suite [14]. They highlight that, although AI has a great potential in detecting the possible threats in large data, currently, the methods of AI-based detection of threats do not offer the required explainability and transparency to build trust, especially concerning the complicated environment of large datasets and the interpretation of the model choice. Their study will help overcome such constraints by making sure that the entire process of the lifecycle of the LLM-based vulnerability detection system is clear and interpretable. The essence of their suggested architecture is a transparency obligation

practice, carefully incorporated in three dimensions: Data Transparency, Model Outcome Transparency and Explainability. Data Transparency also ensures that processes of data collection, data preprocessing (caused by combining classes with dissimilar data) and overall data quality are well documented. Model Outcome Transparency involves accountability with respect to model predictions and model performance, and applies measures such as accuracy, precision, recall, F1-score, ROC AUC and uses such measures to guarantee transparency and the model intelligibility. It has greatly contributed to explainability as being regarded as a strong feature of the work, which is a focus on a unique combination of SHAP, LIME, and attention heatmaps. SHAP offers global and local feature importance, LIME can offer the ability to locally explain specific predictions as a result of perturbation in input data and heatmaps of attention can visually demonstrate the role of various tokens in the model decision hence making the entire process of decision making process more comprehensible. This integrated method was experimentally proved to be effective on the DiverseVul dataset, a large set of C/C++ source codes. Following fine-tuning, the BERT model gave good results in vulnerability detection with 91.89 Accuracy, 87.90 F1-score, 91.99 Precision, 84.16 Recall and ROC AUC of 91.47. The explainability analysis also indicated that the following tokens had a great influence on the predictions of the model: vulnerable, function, "mysql_tmpdir_list", and "strmov". This study does not only contribute to the development of high-accuracy vulnerability detection, but also preludes the establishment of trustful AI solutions achieved through transparency and interpretability of difficult machine learning use cases.

The article by Lin et al.(2022) suggests a new pre-training model to solve the critical problem of encrypted network traffic classification, especially due to the imbalanced and content-invisible nature of such data [8]. Having identified the drawbacks of current solutions that are too dependable on deep features or human-constructed features having inadequate generalization, ET-BERT uses massive unlabeled encrypted traffic to train powerful and generalizable datagram-level encodings. The model converts raw datagrams into language-like tokens, which are bundled into "BURSTs" which denote transmission-guided structures. It subsequently uses a Transformer architecture with two self-supervised pre-training tasks: Masked BURST Model (MBM) which learns contextual relationships between datagram bytes and Same-origin BURST Prediction (SBP) which learns transmission relationships between BURSTs and thus learns traffic-specific patterns using massive unlabeled data. The efficacy and generalization possibility of ET-BERT are comprehensively tested on five encrypted traffic classification problems, such as general application classification, malware classification, VPN traffic, Tor traffic, and the new TLS 1.3 protocol. The experiment outcomes prove that the ET-BERT has a better performance, reaching state-of-the-art results and significantly outperforming the current techniques with high increases in F1-score (e.g., 5.2% improvement in ISCX-VPN-Service, 5.4% improvement in Cross-Platform Android, and 10.0% improvement in CSTNET-TLS 1.3). The interpretative analysis is also offered by the authors, who connect the effective performance of the model with the imperfect randomness of ciphers and demonstrate the strong performance of the model in few-shot learning cases. It can therefore be considered that this work provides a developed, pre-trained system of encrypted traffic classification with high-generalization capabilities and sensitivity

to new encryption methods.

Nouman Ahmad and Changsheng Zhang (2025) in their article discusse the key issue of eliminating black-box behavior of Large Language Models when detecting vulnerabilities in source code [22]. The conventional tools of static analysis are prone to false positives and fail to detect context-specific weaknesses, whereas, despite their claim, the LLMs are not always transparent. The authors propose a resolution to these interpretability issues dependent on A BERT based LLM prompting them to include Explainable AI (XAI) methods, such as SHAP and attention heatmaps, and an end-to-end lifecycle approach to transparency that is a part of the whole-LLM approach. This framework has been constructed in such a manner that auditing and understanding of model decisions can be done which would help build credibility in AI-assisted vulnerability detection. The methodology has been formulated into three steps which are data pre-processing, model test and a transparency framework. Pre-processing of the data includes processing, formatting and cleaning of data, and addressing the imbalance of classes to avoid bias in models. The model evaluation pipeline optimizes a BERT-base-uncased model with tokenized data, Adam optimizer, and early termination. Three aspects, including data lineage and quality assurance, interpretability of model output and explainability of the decision making process, form the basis of the transparency framework. SHAP offers both global and local importance of input features to explanations of predictions and attention heatmaps give a visual representation of important interactions between tokens. Experiments with the DiverseVul dataset show better performance of the proposed method, with a 92.3% detection rate and higher than CodeT5 (89.4%), GPT-3.5 (85.1%), and GPT-4 (88.7%) in the same test condition, with SHAP analysis revealing such predictive tokens as: susceptible, and function. This methodology can be effectively used to provide both high detection and essential model interpretability, which is a viable and strong security analysis approach.

Su and Su (2023)introduce a new framework based on BERT and aimed at detecting malicious URLs with a very high rate of accuracy, which is a critical task in the domain of cybersecurity [12]. The authors emphasize the flaws of the previous deep learning and ensemble models that tend to be inadequate at representing the semantic relationship within the URL string and are not able to work with a wide range of different data types and domains. To surpass these drawbacks, they suggest to use the bidirectional encoder representation and self-attention mechanism of BERT to improve the comprehension of correlations between URL tokens, generated explicitly by URL string or by extracted URL features, and then use a classifier to decide whether a URL is malicious. The effectiveness and flexibility of the BERT-based model were strictly tested on three public datasets (URL strings, URL features and ISCX 2016, both strings and features) and the Internet of Things (IoT) and Domain Name System over HTTPS (DoH) attacks specialized datasets. The system showed incredibly high accuracy levels with a 98.78 percent on Kaggle, 96.71 percent on GitHub and 99.98 percent on ISCX 2016 dataset on binary classification. Such a comprehensive analysis highlights the strength of the model and flexibility in working with all types of data and its potential to be deployed in real-time in different cyberattack schemes to be a great contribution to the online security system by quickly discovering malicious URLs.

Han et al. (2021) suggests a new general framework to overcome the acute lack of interpretability in Deep Learning (DL)-based anomaly detection systems in security applications [5]. Although unsupervised DL models promise to be highly promising in identifying unexpected threats and are highly accurate, their black-box architecture negatively affects adoption, complicates trust-building and finds it hard to identify mistakes in their systems, as well as reducing false positives. The currently used methods of interpretation, which were designed mostly in the context of supervised learning or non-security, are not suitable to the special needs of unsupervised DL in the security setting. DeepAID seeks to address these shortcomings by delivering quality, readable, stable, robust and efficient interpretations to unsupervised Deep Neural Networks (DNNs) in order to increase the viability and reliability of these security systems. DeepAID contains two fundamental methods: Interpreter and Distiller. The Interpreter component models anomaly interpretation as an optimization problem, and seeks a normal reference to bring out deviations as well as considering special constraints in the security domains, including fidelity, conciseness, stability, robustness, and efficiency. In addition to this, the Distiller is a model-based extension, which provides the opportunity to engage the human-in-the-loop interaction, by providing feedback on interpretations, and this information is stored as rules in a two-Finite State Machine (FSM) architecture. Besides aiding in automatically identifying similar anomalies, this mechanism also plays a major role in eliminating false positives. Comprehensive tests have been conducted on three categories of security-related anomaly detectors (tabular, time-series, and graph data) to show that DeepAID is superior to the previous models in terms of fidelity, stability, robustness, and efficiency. The study brings out the ability of DeepAID to assist security operators to interpret the model decisions, diagnose system failures and practically lower the false positives and it ultimately leads to the trustiness of the DL-based anomaly detection.

Molleti et al. (2024) attempts to address the increasing levels of complexity of cyber threats that have rendered traditional defense systems inefficient in combating them in their article [17]. The authors examine the use of the LLM agents to process logs, contextual anomalies content, the derivation of patterns in network traffic, and user behavior analysis and, it could be concluded that it can identify even larger users with a bad motive that is missed by the rule systems. Besides that, the note analyzes the preservation in which the automation of threat management promoted through the use of the LLM agent may contribute to localized skills, e.g., processing, and prioritization of incidents and efforts to analyze a Ryanoclello through LLM agents. These agents can improve situational awareness and even help adjust security policies dynamically. The paper also highlights that integrating LLM agents into existing systems like SIEM and AI-Ops opens new possibilities for developing proactive defenses in fields of security. While the authors acknowledge real challenges like wide attacks, limited comprehensibility, and high resource usage, they remain hopeful. They believe that future LLM agents will evolve to include multi-modal threat analysis and quantum-safe LLM-based cryptographic methods bringing a better result. This paper points out that the potential of the LLM agents in revolutionizing cybersecurity lies in the fact that the software instills greater speed, increased accuracy as well as scalable machine velocity threat detection and response platforms.

The paper by Mohammed (2021) analyzes how artificial intelligence (AI) transforms identity and access management (IAM) through precise and efficient automated user access control procedures [6]. It shows core functionality of IAM and demonstrates AI-based security solutions provide enhanced real-time access decision security by delivering context-based analysis. By using machine learning, IAM detects peculiar user activities which subsequently enhances authentication protocols and lowers manual error frequencies. The research details AI-based identity verification methods that combine biometric data and behavioral analysis to build efficient security frameworks that no longer need conventional passwords. The study explores how IBM uses AI to improve Identity and Access Management (IAM) by connecting adaptive access rules with fraud detection systems to enhance enterprise security. It also shows how AI helps organizations meet rule based standards, improve risk monitoring, and reduce the costs involved in traditional access process operations. IAM technology development will establish complete automation systems enabled by AI-driven behavioral access controls to replace conventional authentication methods and result in cybersecurity changes through enhanced operational efficiency as well as accuracy and adaptability.

Khare & Arora (2024) in their research paper looks to explore how artificial intelligence (AI) and machine learning (ML) can modify the risk based identity verification processes [15]. The main focus of this paper is that AI and ML technologies can highly improve the accuracy, efficiency, and adaptability of identity verification. They help detect fraudulent behavior, perform real-time risk assessments, and introduce advanced authentication methods that strengthen the reliability of user verification. This study used a survey-based research method with a meta-analysis approach, collecting data from AI, ML, and cybersecurity experts to assess the effectiveness and challenges of applying these technologies in identity verification. The researchers examined how algorithms such as supervised learning and deep learning perform in detecting anomalies, preventing fraud, and enabling biometric authentication. Key factors included the accuracy and adaptability of the AI models, deployment challenges like false positives and security issues, and variables influencing risk scoring such as transaction history and behavioral patterns. While the survey provided valuable insights, it relied on a small sample size and subjective responses, and it was not deployed for live testing. The results were analyzed to identify agreement levels, patterns, and areas for improvement. Although the study addressed both technical and ethical aspects of AI-driven identity management, it lacked empirical validation and sector-specific applications, which limited the broader applicability of its findings.

Mohammed (2025) in his paper proposes that integrating artificial intelligence (AI) with Identity and Access Management (IAM) can substantially advance cybersecurity and simplify user authentication systems while improving identity management [3]. According to it, not only can we solve traditional IAM limitations such as inefficiencies in access control, lack of scalability and vulnerability to sophisticated cyber threats, but also increase the quality of common user interactions. The research employs a literature review based methodology which is mainly a qualitative and analytical approach that reviews existing literature on traditional IAM practices and the integration of AI technologies, such as machine learning, behavior analysis, and

risk based authentication. The study looks at using AI for automation, anomaly detection and adaptive decision making in IAM. Furthermore, it stresses the application of AI for real time monitoring, predictive analytics and contextual access management. The authors propose a conceptual model that discusses AI and IAM to bring a more secure, efficient and adaptive system. It points out how AI can be used for automatic authentication of low risk cases, for identification and detection of anomalies of user behavior and for the improvement of regulatory compliance. According to the study, by transforming IAM to a proactive system through AI, it becomes possible to achieve better operational efficiency and security.

Mohammed (2023) investigates how IAM (Identity Access Management Systems) can be enhanced through the application of intelligent authentication [2]. The paper mainly focuses on password management and multi factor authentication within the domain of IAM inorder to improve its usability and effectiveness. The author assesses the drawbacks of conventional IAM processes and suggests unified intelligent authentication as a way to maximize operational effectiveness, enhance security, and simplify user administration. The author mainly relies on literature review of existing research and concepts of IAM to identify the shortcomings in traditional IAM practices such as point solutions, static passwords and through comparative analysis, dives deeper into more advanced approaches such as multi factor authentication and AI-driven systems for improved security, scalability and efficiency. The author highlights the need for leveraging the power of AI into risk based authentication and behavioural analytics in order to streamline the user authentication process through reducing vulnerabilities in password related tasks and ultimately improve security compliance.

2.3 Summary of Key Findings

Recent research indicates that transformer-based models and large language models (LLM) are gaining more and more importance in cybersecurity. Models like BERT and ET-BERT have reached very high levels of accuracy (as high as 99.9) when detecting malicious URLs and encrypted traffic. These findings show that models capable of understanding the context and meaning of language perform far better than earlier feature-based approaches. They are also suitable for real-time applications and can adapt effectively across different domains. A key focus in these studies is on interpretability and trust. Frameworks such as DeepAID and BERT-based vulnerability detectors use methods like SHAP, LIME, and attention visualization to explain their decisions. This helps maintain transparency in AI-driven judgments, allowing human experts to review and validate outcomes with an accuracy rate of over 91 at the same time.

LLMs are also being used for policy automation and compliance auditing. Researchers have increased reasoning and accuracy by 20-30% with tools such as Retrieval-Augmented Generation (RAG) and agent-based systems like GPT-4o, Claude-3.5, and LangChain ReAct. These systems can now automatically check security policies, find weaknesses, and run real-time security audits. Tools like DLMS and GPT-4-based policy analyzers have also improved privacy and data protection,

reaching about 94% F1 score while staying accurate even with encrypted data. However, researchers have found some serious risks in LLMs, such as prompt injection, data leaks, and SQL attacks, with some showing success rates as high as 92%. To reduce these threats, methods like layered checking and anomaly monitoring are being used.

All in all, AI based anomaly detection and identity management systems are currently faster, more precise (90-97) percent and flexible compared to their traditional counterparts. However, there are still difficulties associated with maintaining the transparency, resistance to adversarial attacks and safeguarding user privacy. These are the determinants of developing reliable and safe AI-based cybersecurity systems in the future.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

The implementation of the IAMSecure framework required both well-defined hardware and software environments, along with structured datasets to support machine learning and large language model (LLM) experimentation. The system works on combining locally fine tuned model BERT-tiny and larger language models (Mistral-7B, Gemma3-270M) enforcing the necessity to specific hardware and software requirements. From a hardware perspective, there are two phases one for training/fine-

tuning phase and other for deployment. For the fine-tuning phase, CPUs having Intel Core i7 (10th Gen or newer) or AMD Ryzen 7 (5000 series or newer) worked effectively to handle different data processing and model orchestration. A minimum of 16 GB of DDR5 RAM is required to manage the larger amount of datasets and model checkpoints during fine-tuning by BERT-tiny. NVIDIA GPU (RTX 4060ti with 16 GB VRAM) was necessary to handle batch processing operations efficiently. In the phase of inference, to run the model locally, a consumer sided system with a modern CPU, RAM of 16 GB and a GPU of at least 8 GB of VRAM could be sufficient. Access to Mistral-7B and Gemma-2B will be via API, so a stable internet connection(otherwise a locally downloaded model will be required) with minimal local hardware impact will be needed beyond running client applications.

The project relies on a specific stack of open source software and libraries. For operating system running, Ubuntu 20.04 LTS or similar Linux distribution will do. Python 3.10+ served as a core programming language along with libraries such as Pytorch, Tensorflow, Hugging Face Transformers, Sckit Learn, Natural Language ToolKit for initial processing and tokenization. Django was adopted as the backend framework for deploying IAMSecure and serving the process of pentesting. The data

requirements were equally critical. The dataset size consisted of around 70,000 text samples, labelled with some as benign and some as malicious. Samples are stored in json format. 80% of the data were kept for training purposes to fine-tune the model, and 15% for validation set with 5% of the data for the final, unbiased evaluation testing the model's performance. Together, these datasets provide sufficient scale

for both supervised learning and grounding of LLM based remediation. With the

requirements, it is observed that lightweight supervised models such as BERT-tiny and LightGBM, RNN could be trained on modest hardware with limited resources, but to balance out to get a proper output with integration of LLM demands more advanced GPU hardware and more advanced pipelines. There should be a proper tradeoff to ensure adaptability within available infrastructure. The design adhered to OWASP top mitigation practices types, defending against different types like SQL injection, XSS, CSRF, SSRF and many more. Strong password hashing, CSRF tokens and strict role based access controls were analyzed for giving to the system the solution of maintaining a secure baseline.

3.2 Societal Impact

The project work will have a significatory meaning on society in an effort to keep the privacy of people secure against emerging threats. In the process of connecting to the policies to be saved, detection of malicious scripts and the creation of the actionable advice founded on the same, it can significantly mitigate the chance of infractions committed against data that might compromise the sensitive personal information. A guideline that may be created to ensure economic damages and setbacks are avoided can be constructed by different sectors in the economy. Given that law, it fosters a legal aspect of code diagnosis raising a legal doubt on how the code is automatically analyzed anyway. To take the privacy upkeep maintenance of the sensitive data, the sensitive analysis may be performed through introducing such models in their systems with distinct data handling guidelines. The system provides regulatory framework compliance such as GDPR and HIPAA to verify risk management and access control recommendations. The end result of all these in the society is creating trust, making organizations less vulnerable and performing their operations safely in our digital-dependent world.

3.3 Environmental Impact

The first environmental problem is the energy usage of the training and operation of huge language models. Training BERT-tiny to miniature size and repeated API calls on large models such as Mistral-7B are extremely energy-consuming. A two stage strategy was undertaken to suggest an enduring approach in this regard. The other local models which were of smaller sizes like BERT-tiny were found to be suitable to cover the initial scanning step. The fine-tuning process showed to be running with a minimum number of epochs giving precise output. Following pepper tuning, it was studied later with access to the larger LLM models with modular deployment. With current energy efficient GPUs such as the RTX 3090 they can be used to lower the overall electricity usage as well as an appropriate remedy of working.

3.4 Ethical Issues

This design aspect of the project is highly central since it is ethical. The training set can contain bias whereby the model would vary in its outcome across one pattern of

a programming language to another dissimilar pattern of generating equity intending to develop a fictitious safety. Sensitive information might be a part of scripts under analysis by the system, that ought to be handled so that these data may be safely handled and anonymized in most cases and consented clearly in advance to analysis. Professional responsibility further was exercised in averting the abuse of the system. Indicatively, as the detector recognizes the malicious inputs, guard has been provided on command to ensure the system does not create any malicious inputs. A set of guidelines that stipulate the limitations of such a tool need to be clear to establish the name of accountability regarding the failure of the tool to detect false positives. To mitigate this, a balanced data in training containing sufficient numbers of benigns was employed, and human assessment came into the picture as a workflow. The management of ethical concerns is highly desirable to demonstrate the project's association with the responsible AI that must be transparent and safe to be used by all.

3.5 Project Management Plan

A structured plan was made in completion of the project with division of responsibilities among individuals and completing the task due time. The work process can be divided into several phases: Dataset preparation, baseline model training (LightGBM, RNN/GRU/LSTM), BERT fine-tuning, LLM pipeline integration, and final deployment testing. After researching recent works by reading articles the problem was identified and a suitable dataset was looked for through different sources. Access control dataset containing about 50 features that could be used in analyzing the factors that are essential in IAM field access. From there security measured the weaknesses to the system in building the website by scanning and pentesting. For vulnerable websites, precaution measures are to be pointed out which were given by LLM models. For this purpose, we fine-tuned our BERT-tiny model by introducing several benign and malicious collection samples and then tested the performance towards fulfilling criteria. The pipeline was developed later to integrate to the model with the Mistral-7B and Gemma-2B APIs and end-to-end testing and some refinements made it give suitable results according to how the inputs were given. The recommendations were given based on the type of malicious attacks that the problem caused for through the CLM that was built to the system serving a suited explanatory analysis for the users. Later all the information was gathered and prepared for the final report writing.

From a budget perspective, access to a high performance computing environment or cloud GPU (e.g. AWS EC2 instance) will be required for model training with estimated cost: $30 to $50 for required computing hours. The budget for API calls to Mistral-7B and Gemma-2B could be estimated for around $20. To manage all the resources and maintain transparency, one primary researcher should be allocated responsible for all phases with collaborating to an advisor for guidance. There should be secured storage location kept for the dataset and checkpoints of the model as well as ensuring a parallel progress to adapt to the system in a smooth process.

3.6 Risk Management

Some potential risks in the project were to be pointed out to identify and work on. Firstly, the model performance not being up to the mark in giving the desired accuracy might be a common case and in that case, it is required to work on changing hyperparameters, also augmenting the dataset with cautious checking could make the accuracy to be corrected for further use. To check if the model is lacking in its performance, considerations should be done in taking a slightly larger model of better functionalities.

There could arise risk of API to be unavailable after a time or maybe with a change to the price or limitations. Keeping that in mind, the design should have a modular architecture allowing for easy swapping of backend LLM. The cachings are to be implemented for queries that come in repetition. API usage is to be closely monitored. Lightweight models trained locally before scaling will help in omitting operational risks of handling delays due to limited GPU access.

The dataset collection from public sources could have mislabeled examples, making it unsuitable to expect a good performance from the model. For such, personal gathering of the dataset was made and the data should be validated and revised multiple times with caution.

3.7 Economic Analysis

To make ideas about development and operational costs for this system, it is necessary to estimate the cost-benefit to the system. The development costs primarily involve a budget from the Project Management Plan (around $900) for training BERT and LLMs if performed entirely on cloud GPUs. By using optimized local models and institutional resources, the expenditure could come low. The operational costs could be in the range of $50-$100 per month for a small organization in terms of API calling and maintenance.

The average cost for a data breach incident for a small business can be tens of thousands of dollars. Here, even if the system could prevent a single security incident, it could provide a mass return on investment. The benefit lies in the reduction of the loss in financial terms, reputational damage, and saving operational downtime that are related to cyberattacks. The tools ability to provide clear explanations and ideas to users also saves time for looking after threat analysis, and thus it leads to an increase in productivity for the companies.

Chapter 4

Proposed Methodology

4.1 Design Process or Methodology Overview

A systematic layered approach was followed in the design process to add traditional access control analysis with modern AI detection mechanisms to enable adaptive decision-making processes. The initial phase was to understand security behavior through IAM policy assessments. Several access control related features were analyzed to generate a security score for given inputs, which were then integrated into a web-based application. This application allowed users to test URLs and input configurations, giving insights about the faulty factors in the security posture of the system. This stage acted as a motivation for introducing a more dynamic and intelligent detection framework.

The dual-model pipeline enables the analysis of IAM vulnerabilities while connecting with real world view to visualize proper reasoning. Continuous testing and feedback analysis help in strengthening security access control. Building on this foundation, the study highlights the need for adaptive solutions that can evolve with coming threats, leading to the development of a multi-model pipeline centered around Large Language Models (LLMs). The work follows some quantitative and structured phases of data collection, malicious type introduction, training and validation. For the detection tasks a large dataset containing a mixture of both benign and malicious samples were first prepared for fine tuning with lightweight BERT-tiny as input. Following the step, the categorization of specific attack types is allowed. It produces advanced generative reasoning after proper understanding of types and acting according to that with the help of LLMs such as Mistral-7B and Gemma3-270M. These acted as a support for summarizing the output into meaningful advice.

In this whole process, the model gradually learns and adapts itself to better handling any kind of newer threats to come. The design is structured in such a way that allows proactive threat detection and policy improvement that is understandable to humans. Moreover, it allows technological innovation with realistic cybersecurity requirements. It explores the possibilities of grouping different malicious attacks as well. Besides, the system can execute detection tasks on small local devices while the remediation can be executed on more powerful cloud servers for its architectural design. This modular design supports scalability and makes way to integrate with actual enterprise processes.

4.2 Preliminary Design or Design (Model) Specification

The preliminary design of the project plan was developed in two major phases: IAM based machine learning web application and LLM-driven security evaluation framework. The objective of this design was to develop a computational method of tracking vulnerable factors from IAM security tools and preparing for analyzing to provide a solution based on the type of potential vulnerabilities. To achieve this, several models were chosen and worked on for train-test and preparing for a real time implementation utilizing a web based platform. The complete methodology for the machine learning based IAM phase is given in figure:4.1

4.2.1 Web App Implementation

Data Collection

The dataset titled "Cloud Access Control Parameter Management" (October, 2024) has been obtained from the Kaggle platform in the format of a publicly shared csv file in order to use for this research [16]. Kaggle is a well known public platform where data is curated for various machine learning and data science practices. The dataset used for our thesis contains numerous features which provides an overview about the access control settings, policies and control strategies applied in various cloud services. Thus, the dataset is coherent with IAM, which is the theme of this research.

The original data set comprises 100,000 data points and 51 features. These features are associated with cloud-based IAM configurations and security practices. It has some significant columns that directly correlate with Identity and Access Management (IAM) which consequently play an important role in deciding on the security status of an organization. The main IAM-specific features are Authentication Mechanisms (including MFA, SSO, and password-based login) that define the way identities are verified securely. Authorization Models, such as the RBAC and ABAC identify the approach to assigning access rights to users depending on their roles or attributes[7]. User Identity Management, User Roles, and Access Levels are the core features that define the level of individual access and authorization in any organization. The Access Revocation, Privileged Access Management, and Least Privilege Principle guarantee that the access is provided correctly and is removed when it is not required, as it minimizes risk. Session Management, Third-Party integrations, Federated Identity, Audit Trails are other great features that work collectively to facilitate secure, traceable and policy oriented IAM activities.

Additionally, the dataset includes the target variable Security Score for each configuration/row that reflects the overall security strength following IAM strategies. The target column has the following classes (2,3,4,5) where higher values of score represent stronger security measures and vice versa. All together, these features provide a strong basis for analysing the strength of IAM policies adopted in any multidimensional organization.

Figure 4.1: Methodology Flowchart

The flowchart in figure 4.1 shows the sequential processes starting from data collection to splitting for training and testing with preparing for model measures is shown. The detailed description is being discussed in the later section.

Data preprocessing and Analysis

Data preprocessing is an important step in any research, especially crucial in case of security related tasks to ensure the data integrity for overall performance of machine learning models. An overview of the strategies adopted in order to make the dataset more suitable for model training has been given below:

Feature Encoding:

Among the features, 7 columns include object type variables (categorical/textual) and the rest contain boolean type variables(True/False). In order to apply machine learning algorithms to categorical data, the most important features Authentication Mechanisms, Authorization Models, Access Levels, User Roles, Cloud Service Provider, Geolocation Restrictions, and Data Sensitivity Classification were encoded by using Label Encoding. In this approach each unique category is masked with an unique integer so that the data can be interpreted by the models without rendering itself to high dimension. For example, in the Authentication Mechanisms column the values Password, MFA, SSO and Biotic were coded as 3, 2, 1 and 0 respectively. Correspondingly, the categories of User Roles were associated with the values 0, 1, 2, and 3 for Admin, Developer, Guest, and Manager respectively. This part of data pre-processing is especially important for computability and interpretability of different categories of any feature inside a machine learning model.

Boolean Conversion and Cleaning:

After an initial review of the dataset, it was found that many features contained Boolean values (True or False). These were systematically converted into binary form, with 1 representing True and 0 representing False after processing. This step ensured consistency in the dataset and allowed numerical computations to be carried out efficiently. It also helped machine learning models such as tree-based classifiers and support vector machines process the input data without type errors or inconsistencies.

To make the model easier to understand and more accurate, data was analyzed with Security Scores of 3 and 4. The very high (5) and very low (2) scores were extremely less in number and could make the results confusing. By focusing on scores 3 and 4, we got a balanced and fair comparison between medium and high security levels, which helped the model work better.

Feature Engineering:

In order to enable a more comprehensible interpretation of IAM security status, the raw numerical scores of the Security Score (2, 3, 4, 5) were translated into broader qualitative risk types. This conversion was done using three engineered features: **Smart Risk Score**, **True Count**, and **Risk Admin No MFA** which are derived from the existing feature set. True count is the total number of activated security features denoted by positive boolean value, Risk Admin No MFA denotes users not having multifactor authentication (highest security) in spite of having higher privilege over the system and lastly Smart Risk Score is a feature developed using a custom risk scoring formula that penalizes critical features not being enabled. This approach did not only improve the capacity of the model to identify the underlying risk factors but also opened the concept of domain-specific intelligence in the model.

Handling Class Imbalance:

When analyzing the IAM dataset, the class imbalance in the Security Score variable was recognized as a crucial issue. Although the most infrequent classes (class 2 and 5) were dropped, the skewness of the target variable persisted. To mitigate this a normalization approach aimed at reducing the possible bias was adopted which included clustering the label class 4 further into 3 groups (40, 41, 42) using K-means clustering and later reclassifying the security score values into three new groups: low, medium, and high. The class distribution from before and after processing the dataset is shown in figure 4.2. The objective for this conversion was to attain an overall balanced distribution of the target variable. This strategy promotes fairness and effectiveness of supervised learning models by avoiding dominance of one particular class.

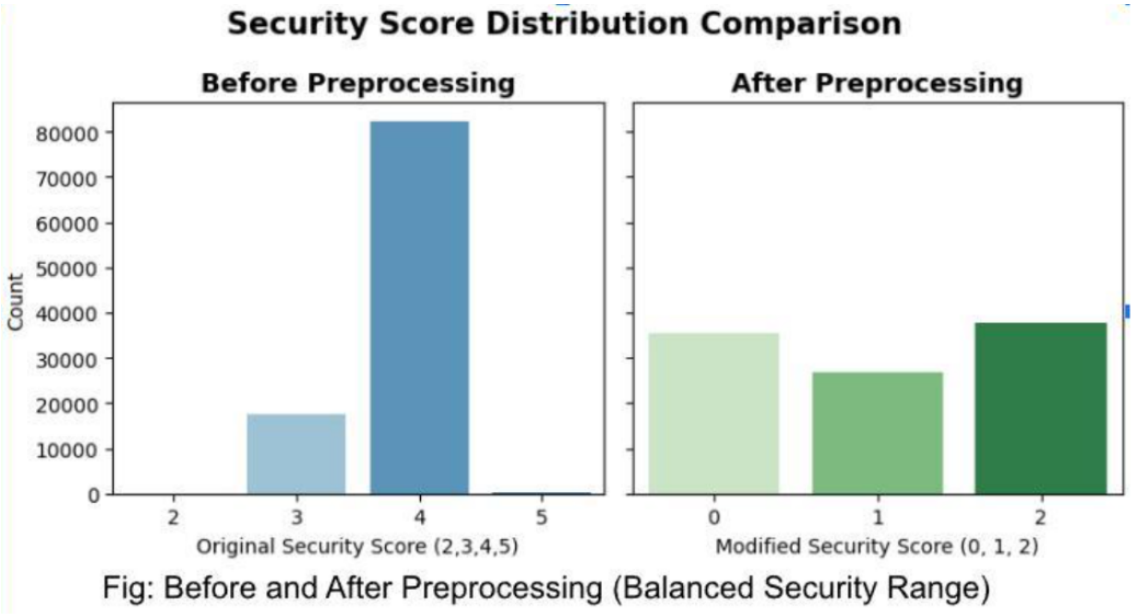


Figure 4.2: Security Score Class Distribution

The algorithm recurrently scanned the range of bounding possibilities with lower and upper thresholds and found the combination, which resulted in minimum variance of the classes. In particular, it was assumed that observations smaller than the lower threshold (smart risk score <= 14) were low, those between the thresholds medium, and those above the upper threshold (smart risk score > 16) high. This re-categorization trained the initial numeric target to produce a well-balanced categorical variable that fit into the multi-class classification.The cleaned data with all the re-categorized labels were saved as a CSV file to use during the next training and testing phases of the model.

Model Architectures and Training

This section presents the implementation of multiple machine learning models designed to evaluate the security strength of a system based on its IAM (Identity and Access Management) configuration. The goal is to predict the Security Score categorized as low, medium, or high in order to assess how vulnerable or protected a given

policy setup is. After preprocessing the dataset and encoding categorical features, we trained and evaluated multiple classifiers, storing both performance metrics and confusion matrices. Among all the trained models, the best performing ones have been discussed.

XGBoost

XGBoost (Extreme Gradient Boosting) model is an excellent ensemble model which computes and accumulates a series of decision trees in order to minimise errors most efficiently. This model was used in classifying security scores, and the aim was to use its effective gradient boosting structure to deal with complex decision boundaries as well as feature interactions in data associated with IAM such as this one. XGBoost also works very well on structured categorical data such as the access control dataset. Having encoded categorical features and mapped the labels of classes, the model was trained on the balanced dataset and demonstrated an impressive accuracy on the test set of 89%. It also got a high result in all three classes resulting in an overall F1-score of 0.887. As demonstrated by the confusion matrix, Xgboost had perfect classification of all three classes - low, medium and high. In general, XGBoost did a fine job processing the resampled, encoded data to provide a well generalized model which can reasonably predict the level of vulnerability of any system.

LightGBM

LightGBM is a fast scalable library that implements gradient boosting. It is faster than the traditional boosting techniques because of its leaf-wise tree development and its use of histogram-based techniques and competitive accuracy. Because this model processes categorical data and can train fast on large datasets, this model was specifically appropriate to our IAM dataset. It registered an F1- score of 0.809. LightGBM, whose training set was resampled and label encoded where the class imbalance problem was addressed through feature engineering and clustering, achieved an overall test set accuracy of 83%. The model showed perfect classification of the three classes similar to the performance of Xgboost . All in all, LightGBM was a successful multi-class classifier in the application of IAM-based security scoring, where it provided an optimal balance between the accuracy and the computational time requirements.

Catboost

CatBoost, a gradient boosting algorithm, is natively capable of handling categorical features, which makes it a suitable algorithm to use with IAM datasets that contain such variables as User Roles, Authentication Mechanisms,Cloud Service Provider etc. CatBoost was selected in this study due to the capability to analyze complex relationships in structured data with the minimal preprocessing. The accuracy (0.986) and F1 score (0.985) of this model was very good and demonstrated a higher quality of classification. The confusion matrix demonstrated that the precision of the result was high in each category such that there were 6,980, 5,227, and 7,505 correct

predictions in Low-risk, Medium-risk, and High-risk respectively. Few Medium-risk records had been falsely labeled proving that the model is capable of discerning small shifts in access control settings. The combination of the good computational qualities coupled with its interpretability entitles it to security-oriented applications in an IAM system.

Keras Neural Network

The neural network has been trained by Keras as a multi-layer perceptron that evaluated the quality of modeling complex feature interactions in IAM through deep learning. It did the best overall and its F1-score 0.992. Normalized IAM characteristics and one-hot-encoded security score were used to train the model with two hidden layers (128 and 64 neurons respectively) and dropout to regularize the model. It also registered near ideal learning depending on the confusion table with a 0.992 accuracy, precision and recall. The fact that validation and training loss were decreasing throughout the 15 epochs demonstrated that there were no poor generalization characteristics, and there were no signs of overfitting. The number of errors was rather small (e.g. 115 misclassifications class 0, 28 class 1 and 9 class 2). This is a great indication that the model did a great performance in obtaining both linear and non-linear dependency in the IAM feature, a fact which makes the model extremely viable and applicable in real-life systems, where the systems are expected to be accurate in every possible way.

QDA

The Quadratic Discriminant Analysis is a kind of probabilistic guiding model which makes an assumption of normal distributing features in each class. It works quite well in interpreting from a separated variance structure based on Bayes theorem. For the training model to have class-specific matrices of low, medium, high risk results, QDA model was assumed to capture the non-linear boundaries well. QDA has overpowered many more of our data train models with 0.8 F1 score. Our model train test results show that it could be valuable for insights, though the data per class actions in further imbalances, there was a risk for it to fail miserably.

Performance Comparison: Top 5 IAM Risk Scoring Models

Model	Train Accuracy	Test Accuracy	Precision	Recall	F1-Score
Keras Neural Net	0.9998	0.9992	1.0	1.0	1.0
CatBoost	0.9994	0.986	0.98	0.99	0.99
XGBoost	0.9754	0.8955	0.89	0.89	0.89
LightGBM	0.8743	0.828	0.82	0.81	0.81
QDA	0.817	0.809	0.8	0.799	0.8

Figure 4.3: Performance Comparison

Figure 4.3 shows a significant good result for models like XGBoost, Catboost and Keras Neural Net with accuracy reaching almost perfection 97.54, 99.94 and 99.98 respectively. Based on such results of accuracy, precision, recall and f1 score for the models, the corresponding confusion matrices are structured below.

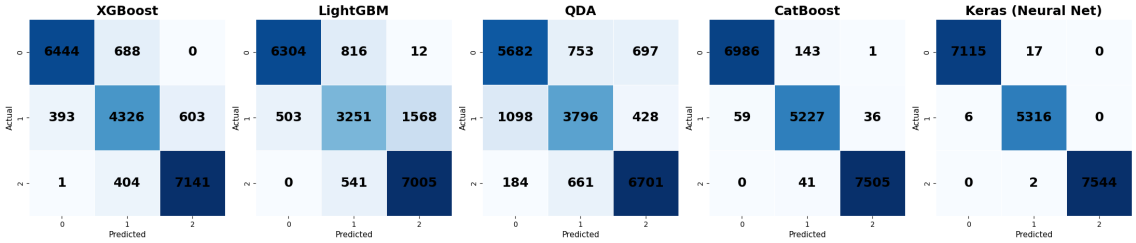


Figure 4.4: Confusion Matrix

It is evident from Figure 4.3 and Figure 4.4 that the balance in achieving accuracy is contained with all the models and XGBoost is selected as the best suited model considering a proper trade-off between accuracy, speed, and robustness in handling complex data patterns.

Model Insights and Framework Proposal

The preliminary establishment of the IAM assessment tool was fundamentalized into an XGBoost-trained machine learning-integrated Django-based web design framework chosen to be the perfect-suited model.

Model Analysis and Specification: Apart from the multiclass classification report aimed at identifying weak spots, a varied group of models- from traditional approach to deep learning train and tests were performed to prepare it for the real-time environment. The purpose was to minimize data overfitting, which could be a common challenge in cybersecurity logs. Random forest, extra trees, and K-Nearest Neighbors, despite receiving a full trained score, the test accuracy dropped by over 0.25. This proved the underperformance tendency of these models when fed any unseen IAM setup.

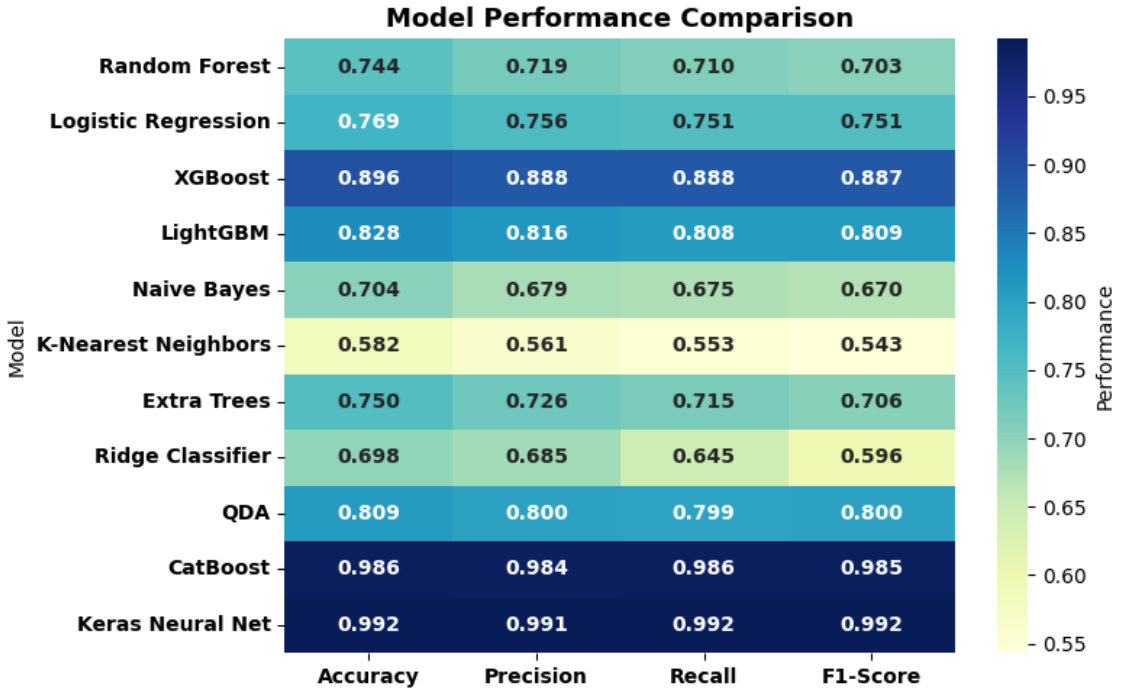


Figure 4.5: Model Performance

On the other hand, XGBoost and CatBoost outperformed with satisfactory results on both train and test nature from the other models from Figure 4.5. XGBoost gave a minimal train-test variance scoring 89.55% on test. The working principle of extensive hyperparameter tuning strategy and faster approach of working on large datasets with L1 (alpha) and L2 (lambda) results in fulfilling high-dimensional security detection related purposes precisely. The potential risk scores were identified with its high resistance to noise functionality making it resilient to complex workloads. CatBoost even outperformed all other comparative analysis, showing excellence in 98.60% test perfection, closer to matching the network baseline, while not further diving into deep learning. The handling technique of the CatBoost model did best with minimal preprocessing reading the categorical and boolean features.

However, in order to deploy to the website, the requirement was to simplify the complexity overall, which XGBoost delivered a faster training time in the preprocessed dataset. The lightweight web deployment acknowledges more explainability to serve in responding to compliances. XGBoost Classifier (iam security model.joblib) and encoders (iam encoders.joblib) was loaded to the model after training process enabling the assess function to collect from user and web inputs and generating recommendation techniques (e.g. enabling MFA for admins).

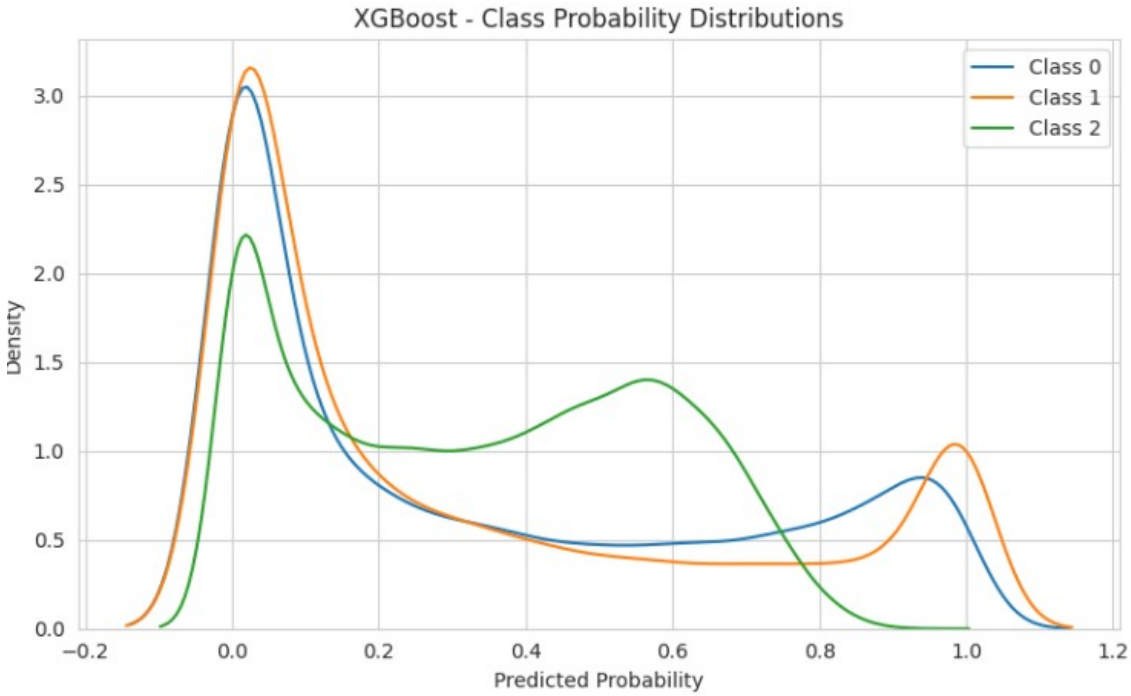


Figure 4.6: Class Probability Distribution: Xgboost

From Figure 4.6, it is seen that the well driven balance between generalising and handling technicalities serves the iterative assessment steps in the Django application plan, shaped by considering XGBoost as the best alignment as a real time visionary tool. [4]

Web Application Architecture :

The robust Model-View-Template (MVT) Pattern for the Django web application framework offered modularity with QuickAssessmentForm allowing user insights validation. Later, the user inputs were merged with default policies, calling assess IAM policy, and outputs detailed result reports based on internal mechanisms as trained towards. It strategizes and defines endpoints with urls(py), with enriching modularity configuring the ScannerConfig based on the established framework.

The enlisted features gave a considerable amount of ideas about the IAM assessment tools feature invigilation. From the larger amount of necessary factors, around five to six critical recommended items were chosen to communicate with the users about their insights and rebuild a correlation view working as a default value in the backend. The efficiency enhancement takes a more better approach with layer based work procedures.

The XGBoost model was utilized as the first layer of working mechanism to give a correct base to the recommendation engine. The security standard based recommendation will not point the vulnerabilities straightly rather it will suggest the upgrading techniques. Based on the framework, A SLM (Short-Language-Model)

will be built as a chatbot for the companies based on the Django framework soundation. The company security standard ideas can be known well enough to guide and reshape towards through the user friendly seamless and low risk model. The higher risk security scores will give a frontier knowledge to various systems and build a stronger safety to minimize cyber attacks.

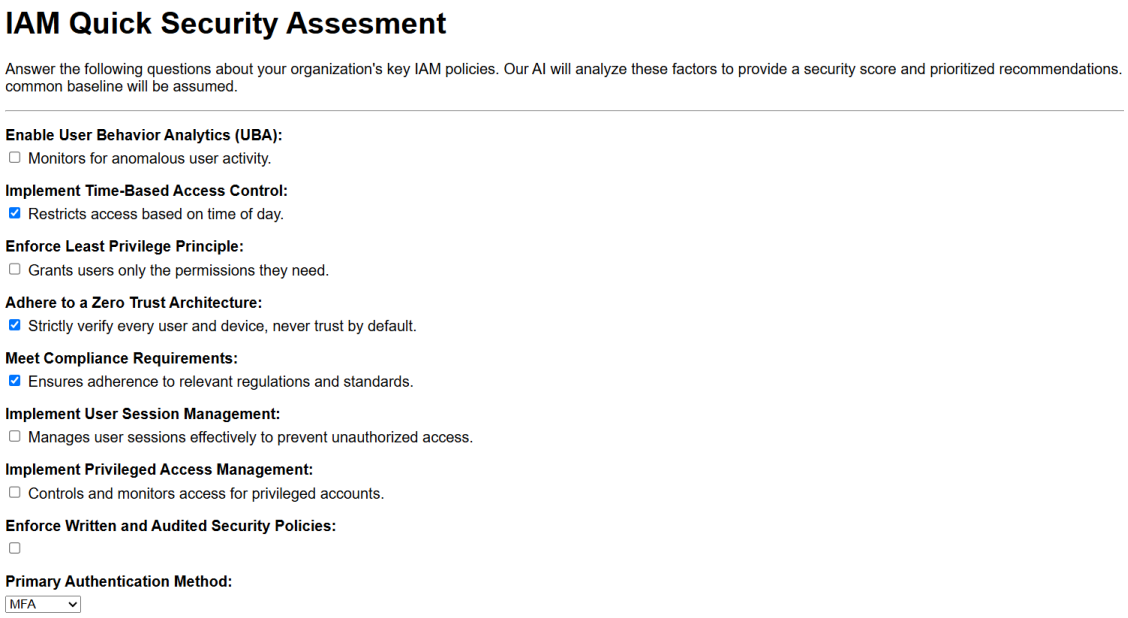


Figure 4.7: Web App Implementation

The IAMSecure tool assesses (Figure: 4.7) with taken user inputs alongside its trained ideas for security issues and detected multiple vulnerabilities, including broken access control and security misconfigurations. It analyzed the website and gave a final security score if low, medium or high.

Limitations

Although the dataset is well defined in terms of access control and identity features, they have certain issues that affect our analysis and the performance of our model. For example, there was a severe imbalance in the categories included in the Security Score label as most of the values were in one of the classes (e.g., score 4). This bias affected the effectiveness of classification models and required resampling or relabeling approaches carefully to train in a balanced manner. There were also numerous features in the form of booleans (e.g. True, False) which had to be preprocessed (at great scale), encoded, and made consistent. Another challenge is that there are some

features whose values are uninterpretable. Fields such as "Authorization Models" or "Access Levels" lack standard definition or domain-specific mapping, which brings a challenge to know what the correct security implications of such values would be without assumptions. Furthermore, the data set does not include contextual data such as user behavior, time-based access patterns, or historical attack logs, limiting its ability to use it for real-time risk scoring. Finally, the evaluation of association

or dependencies among features is limited due to lack of metadata (e.g., feature source, timestamps, or correlation with real-world incidents), causing models to rely on static indicators at the surface level. Improving these shortcomings would make the dataset a valuable asset to solid IAM threat modeling and security assessments using AI.

4.2.2 LLM-Based Security Protocol Optimization

Data Collection

The data set was prepared by combining samples from various sources both benign and malicious. From github Seclist repositories(Danielmiessler, 2025) and Payload List of all the Things(Swisskyrepo, 2025), different types of malware samples like XSS, SSRF, SQL Injections were combined and formed [28] [31]. The final train set sample was composed of around 67,000 samples from which around 40,000 were of malicious types. The type listing of different types of malicious items was taken from github repository and thus the data set was ready for finetuning by different models of LLM.

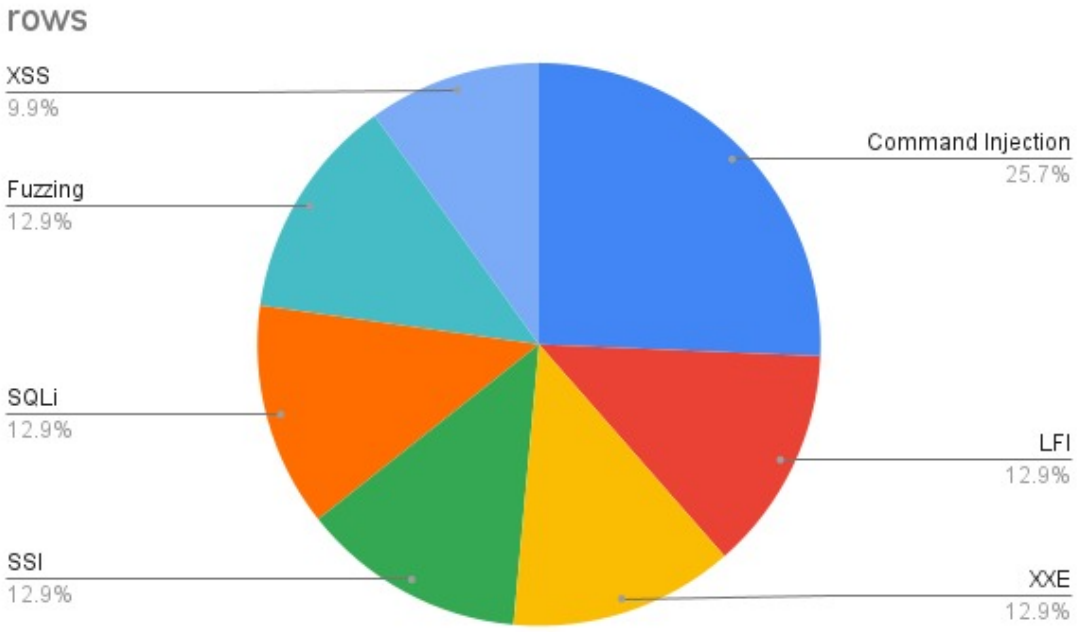


Figure 4.8: Malicious type division

A distribution of malicious samples of various types can be observed in the dataset as shown in the pie diagram in Figure 4.8. The dataset is varying with Command Injection dominating at 9,950 rows, while XSS is the smallest class with 3,848 rows. Mid-size classes like LFI, XXE, SSI, SQLi, and Fuzzing range around 4,990–5,000 rows, with overall even distribution that was considered to be ready for later experimentation.

Model Architecture and Parameter Comparison

Different sets of models were used to tune and use a few shots technique for ensuring a better explanatory analysis given by the Large Language Models (LLMs). The complete workflow for this phase of the research has been given in figure: 4.9.

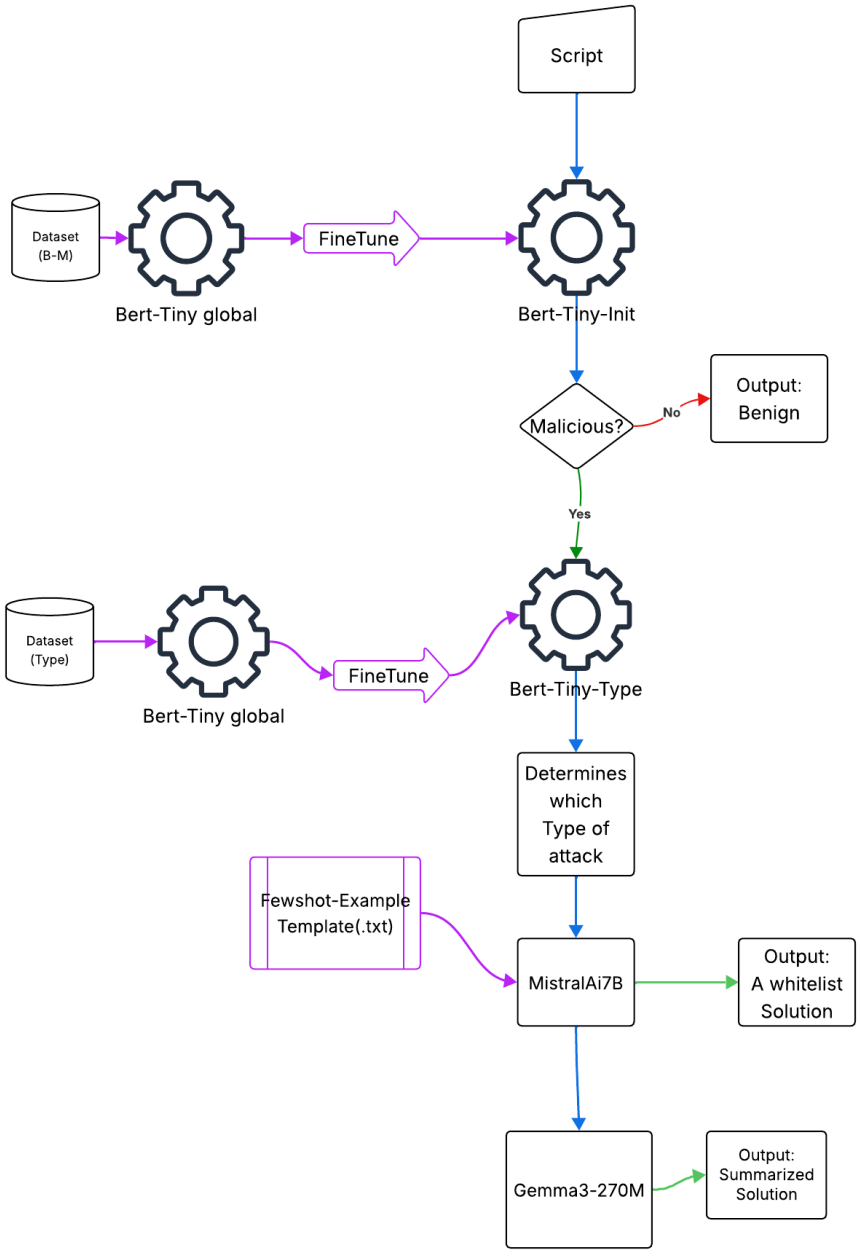


Figure 4.9: LLM workflow

BERT-tiny (Classification):

In this workflow, the Bert-Tiny model is used twice, firstly, as BERT-Tiny-Init and, secondly, as BERT-Tiny-Type. Bert-Tiny is a small implementation of the original BERT (Bidirectional Encoder Representations from Transformers) architecture, particularly designed to achieve computational efficiency without much loss of its strong language understanding properties. It is architecturally represented by one multi-layer bidirectional Transformer encoder which allows it to encode linguistic elements with respect to all the other tokens in a given sequence, and hence creates rich and contextualized embeddings. Due to its low level parameters, it resulted in shorter inference times and less memory, which is essential in real-time applications. The BERT model was trained with OMEGA-9 configuration for several processes of finetuning with changing parameters.

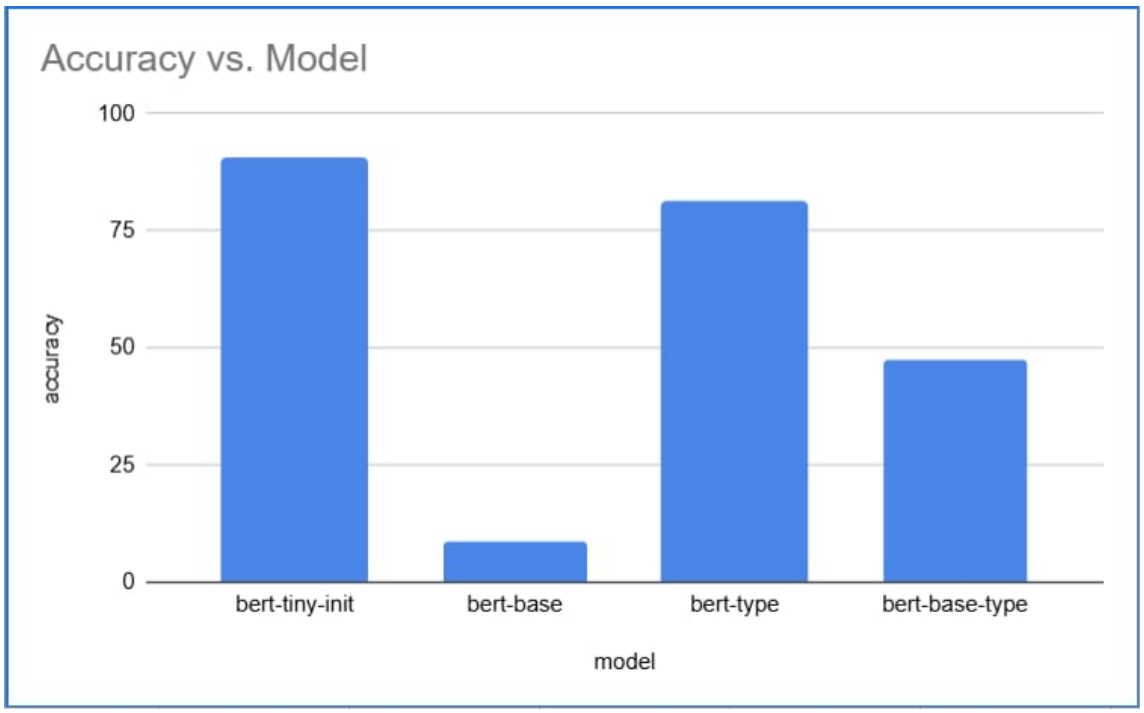


Figure 4.10: BERT model comparison

The action of fine-tuning process on a large dataset (70K) to classify types has made use of the ability of BERT-Tiny to detect complex patterns in text sequences thus making it very applicable in classification that is related to security such as the ability to detect possible threats within a code or command. The initial screening is made fast. On being judged malicious, a script is passed on to Bert-Tiny-Type, which was trained on a type dataset. This is a specific Bert-Tiny case which deals with the type of attack. Such a two-way usage of Bert-Tiny is another example of a two-layered method of threat analysis; a generalized classification is narrowed down by a more specific one, and in both cases, it is effective and efficient with the power of text encoding features of the model.Figure 4.9 shows the performance evaluation on different models of BERT

Mistral-7B (Analysis):

Additionally, in the diagram, MistralAI-7B refers to a 7-billion parameter model of language developed by Mistral AI, following the transformer line of architecture. Like other large language models (LLMs), it is designed based on the idea of a decoder-only transformer stack, which provides it with the ability to produce coherent and context-relevant text when reacting to input prompts. The architectural innovations that are often considered in this model to improve efficiency and performance include Grouped-Query Attention (GQA) to make inference faster, and Sliding Window Attention (SWA) to process longer sequences with reduced computational overhead. Since it has large parameters (7B), MistralAI-7B has significant abilities in understanding complex instructions, being able to produce creative textual content, and perform a wide range of other natural language processing tasks.

In the presented workflow, the MistralAI-7B is involved after the detection of an attack type by the BERT-Tiny-Type model. In its input, it takes the identified type of attack and a Fewshot-Example Template. This structure will make the best use of the generative and reasoning capabilities of MistralAI-7B. The model is triggered into creating a whitelist Solution meaning the creation of a particular countermeasure or mitigation strategy that is well-crafted to the stated threat. The presence of a few-shot template gives demonstrative examples or principles to follow, which allows the model to develop highly applicable and structured solutions, even with limited direct and extensive training on this specific task. Its main force is the ability to integrate various information and create useful advice or executable code that places it as a priceless element in the process of automating incident response generation on a cybersecurity platform.

This LLM analyzer suggests a secure fix to the problem analysis. For example, in pipeline script, it takes a suspicious text (like SQL Query or Code Snippet) like "SELECT * FROM products WHERE category = " OR '1'='1' –"

Once identified as malicious, it gives explanations of malicious reasoning and how to adjust upon such vulnerabilities as given in Figure 4.11.

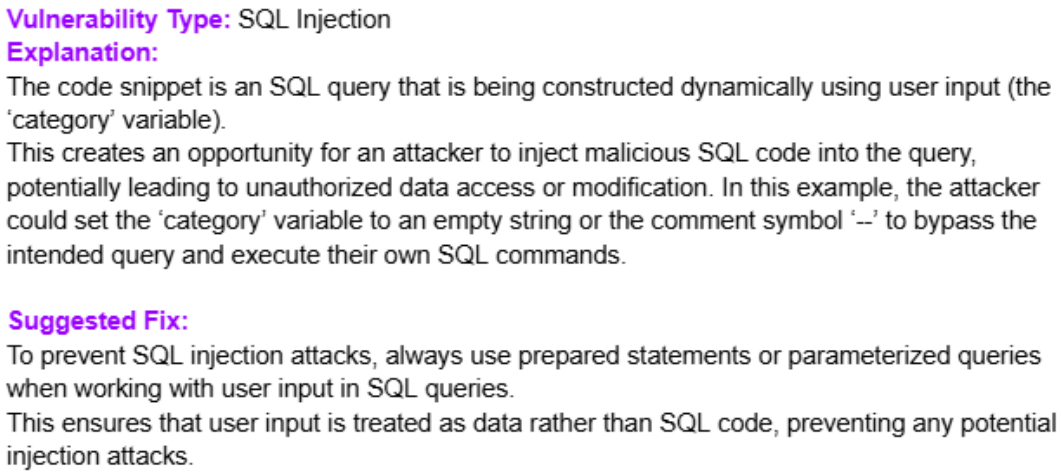


Figure 4.11: Mistral work process

Gemma3-270M:

Gemma3-270M is a compact yet highly capable language model developed by Google, designed for efficiency and deployment in resource-constrained environments. Like its predecessors, it follows a transformer-based, decoder-only architecture with approximately 270 million parameters, featuring fewer layers, smaller hidden dimensions, and optimized attention mechanisms to reduce both memory and computational requirements. Despite its smaller size, Gemma3-270M retains strong reasoning and summarization abilities, making it suitable for real-time applications and on-device inference where responsiveness and low latency are critical.

In this study, Gemma3-270M serves as the final summarization layer in the LLM-driven policy recommendation pipeline. After Mistral-7B generates an extensive remediation or compliance solution, the output is passed to Gemma3-270M, which refines and condenses the content into a concise and human-readable summary labeled 'Output: Summarized Solution." This function is crucial for dashboard and reporting modules, where administrators and decision-makers require clear, actionable insights rather than lengthy technical explanations. By distilling Mistral's detailed recommendations into short, coherent statements, Gemma3-270M improves usability, speeds interpretation, and ensures that complex security intelligence can be quickly understood and acted upon. Its small footprint and fast inference capabilities make it an ideal choice for integrated or edge-deployed cybersecurity systems, balancing precision, interpretability, and efficiency.

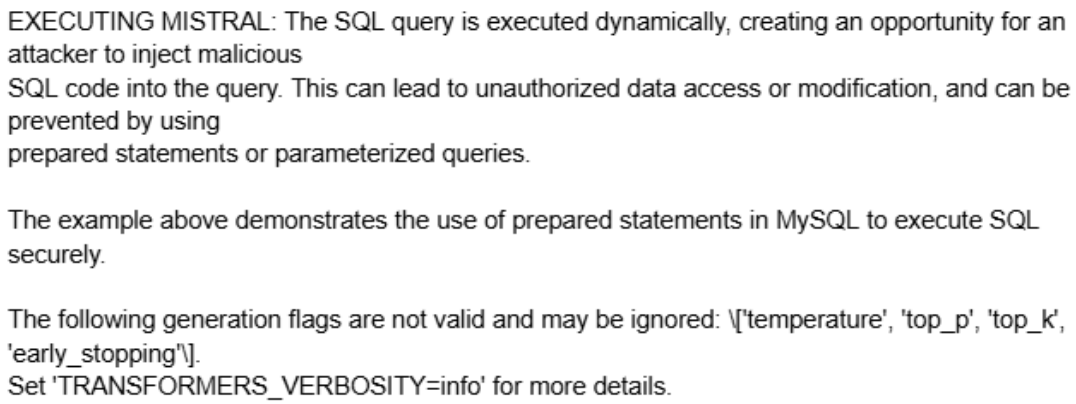


Figure 4.12: Gemma3 example scenerio in the framework

For the same SQL injection described above, Gemma (from Figure 4.11) has provided a detailed summarization to Mistral's SQL Injection findings, emphasizing the prevention method using prepared or parameterized queries.

To design an efficient system, it was important to compare available transformer-based models in terms of parameter size and cost of computation. Since the work was done under a limited hardware system , model selection was a very essential factor hence. From several pre-trained transformer models like BERT, RoBERTa, ELECTRAA, DeBERTa and others, the small sized model BERT-tiny with only 4.4 million parameters, acts as an efficient alternative. After the primary detection is served with a first level filter, it performs classification of identifying whether the dataset is malicious or benign. We chose the one with lowest parameters for maximum optimization.

The working process is described in Figure 4.12. If the script is identified as malicious, it proceeds to the next stage of work with determining the type with the malicious type datasets from which it can properly get habituated with the types of attacks that could be caused. Going in depth with api connection from BERT-tiny type, Mistral-7B was used for suggesting remedies adding a whitelist of mitigation techniques that could be applied to neutralize the threats.

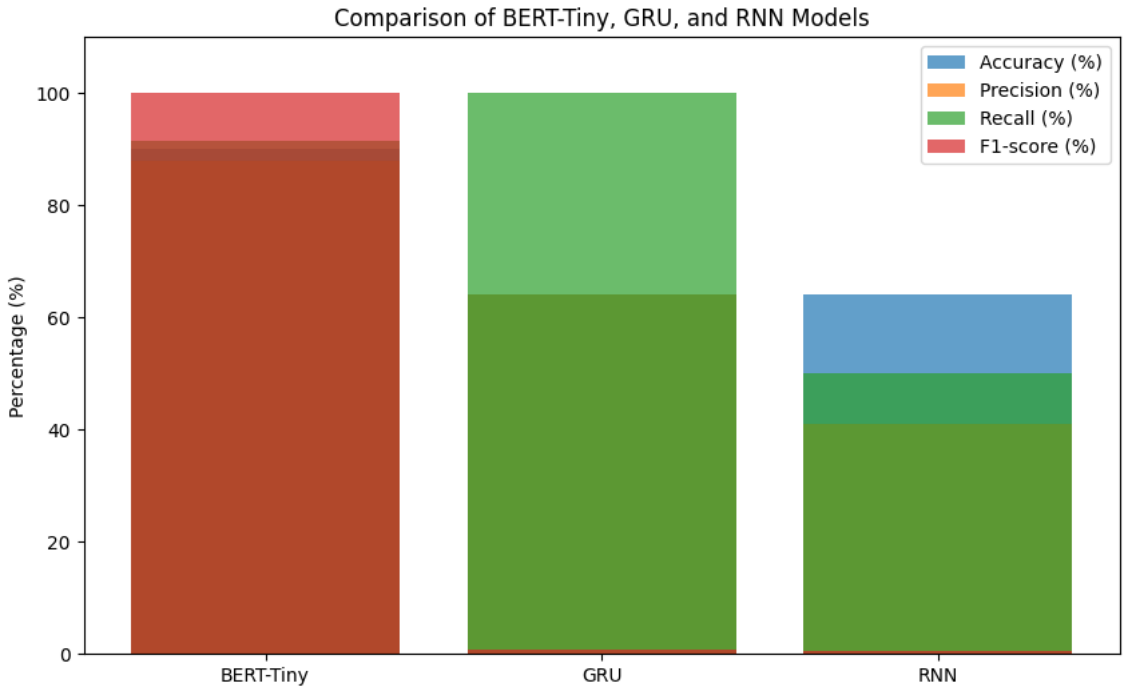


Figure 4.13: Comparison of BERT-Tiny, GRU, and RNN Models

To compare with simple sequential models like GRU, RNN, the data train set was tested and result showed that the GRU model only reaches 64.23% test accuracy, with class imbalance evident: it correctly identifies all class 0 samples but almost completely fails on class 1, reflected by extremely low F1-score (0.01) for that class. Similarly, the RNN model also achieves 64.12% accuracy, with poor precision and recall for the minority class, indicating that sequential models struggle with imbalanced or multi-class attack detection.

Overall, BERT-Tiny's transformer-based approach captures complex patterns in the dataset more effectively, handles class imbalance better, and provides far superior predictive performance compared to traditional sequential models like GRU and RNN shown in Figure 4.13.

In the end, a concise summary was required for users to understand the steps and situation. For this, Gemma-3 was used to generate a human understandable summarized version. This multi-stage pipeline gives the briefing of the system very efficiently in a user friendly format ensuring better allocation of resources.

Parameter Comparison

To ensure that the models would be able to achieve the best accuracy and generalization possible under a certain amount of efficiency, a comprehensive parameter optimization process was performed for all the components of the system and specifically for the BERT-Tiny and BERT-Tiny-Type, as well as the Omega-9 fine-tuned versions. The tuning process involved trials on a wide range of hyperparameters such as learning rate, weight decay, really Vera training, warm-up several, etc. keeping in sight the balance among the model to converge, prevent overfitting and lessen

computation cost. Across several runs, the baseline of the BERT model Tiny (4.4M parameters), proved to be pretty good with over 90% accuracy even in shorter cycles of training. And as different hyperparameter settings were changed, some significant trends were shown. Increasing the number of epochs from 3 to 30 boosted its accuracy from 89.52% to 91.12% as it showed that while training longer there was the ability of the model to adapt to the domain-specific data related to cyber security. Similarly, a learning rate of 1e-4 with a weight decay of 0.01 always returned the best range between being stable and learning efficiently. Warm-up steps around 300 were found optimal to get a progressive introduction of learning, without loss spikes at the beginning of training. These resulted in the basis of the Omega-9 configuration that had the best performance with accuracy of 87.84%, precision of 91.53% and recall of 100% which was best as a balanced and stable model configuration to be further integrated into a model.

For BERT-Tiny-Type (attack classifier) and B-Malicious binary detector, a similar grid search methodology was used to test the influence that the variation of parameters had on the robustness of classification based on multiple attack patterns. Testing various combinations of both factors, namely learning rates between 1e-6 and 1e-3 and weight decays between 0.0001 and 0.05 showed that many configurations underfit or resulted in unstable training, while relatively moderate configurations, near to Omega-9 configurations, gave some consistent and interpretable results. The best performing set up for this model reached 82.45% accuracy and 83.16% for precision after 50 epochs thereby confirming that somewhat longer cycles of fine tuning improved the sensitivity to the context while not impacting the computational efficiency. In comparison to heavier models, like BERT-Base(110M), RoBERTa-Base(125M), and DeBERTa-V3(184M), the optimized trim versions of BERT provided quite magical results with a fraction of resource costs, making them perfect for integrating real-time with the Mistral 7B and Gemma components in the final LLM-driven remediation framework. This optimization process also served as a validation study to obtain near state-of-the-art performance on cybersecurity detection tasks by fine-tuning smaller models with a set of efficient transformers using the right balance of parameters and able to be deployed in lightweight web-based applications.

Chapter 5

Performance Evaluation and Analysis

5.1 Performance Evaluation

The performance of the proposed IAM framework has been evaluated through a combination of qualitative, as well as, quantitative metrics for the integrated LLM model. Using this approach for assessment has enabled us to evaluate not only the raw predictive performance, but also the interpretability, efficiency and reliability of the whole pipeline.

A number of assessment criteria were put in place for this purpose : Accuracy was regarded as the total percent of the correct predictions of the BERT-tiny classifier, whereas, precision and recall measured the trade-off between minimum false alarms and maximum coverage of malicious detection. Efficiency was defined as the sum of time taken from when a suspicious script was typed into the system until the administrator received the final summarized output. Reliability, however, was evaluated based on the consistency of solution to give stable and correct results under a wide range of conditions.

In order to implement these measures, the BERT-tiny model was run on an unseen dataset of 70,000 samples, making sure that the outcome was representative of the model generalization ability outside of the training dataset. Besides this large scale testing, the whole end-to-end pipeline was also tested using a curated collection of 50 scripts, including 25 benign and 25 malicious samples of different classes, like SQL injection, cross-site scripting and obfuscated payloads.

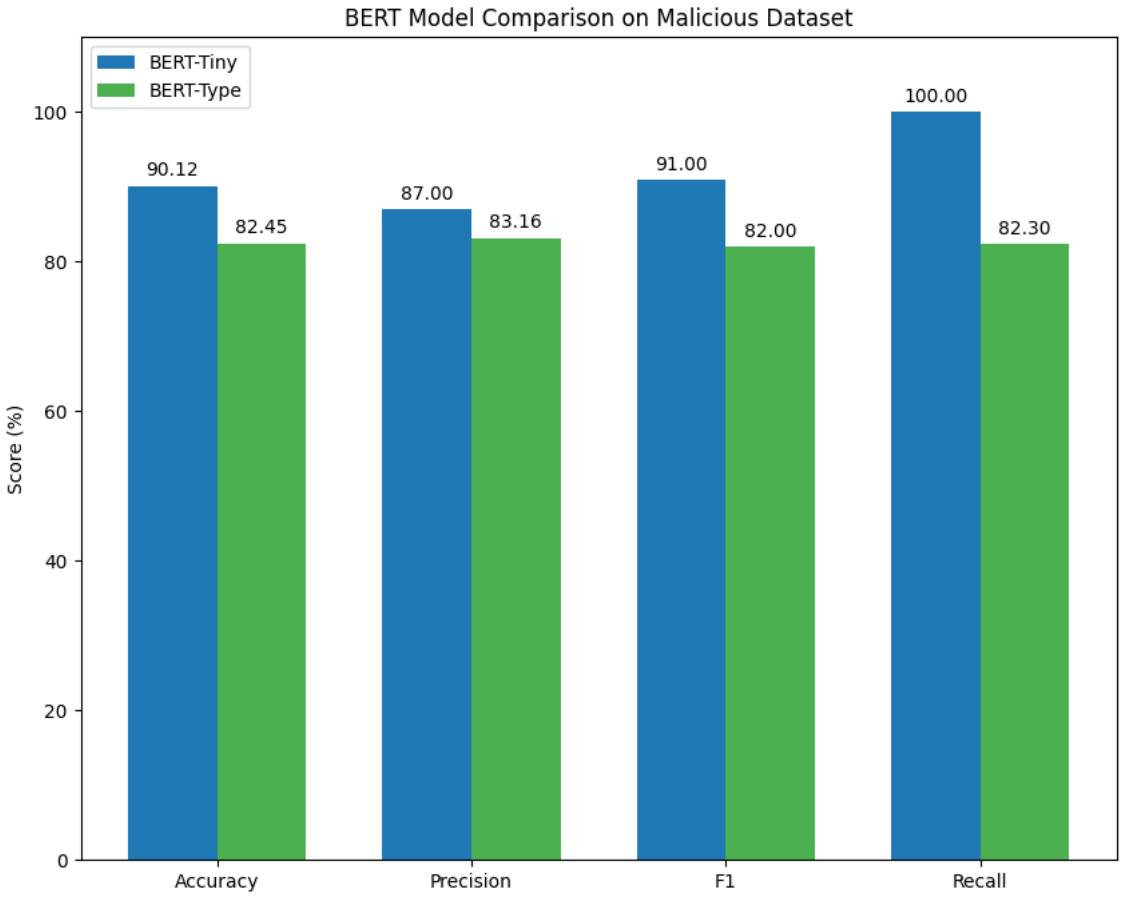


Figure 5.1: BERT tiny init vs type

The results showed that the BERT-tiny model worked really well. It had an accuracy of 90.12%, precision score of 87.84%, and 100% recall. This means it could correctly find almost all malicious scripts with very few mistakes. Each script took about 12 seconds to check, which is fine for testing in groups but a bit slow for real-time use. Overall, this shows our IAM system can be reliable, and ready to be used in real security situations.

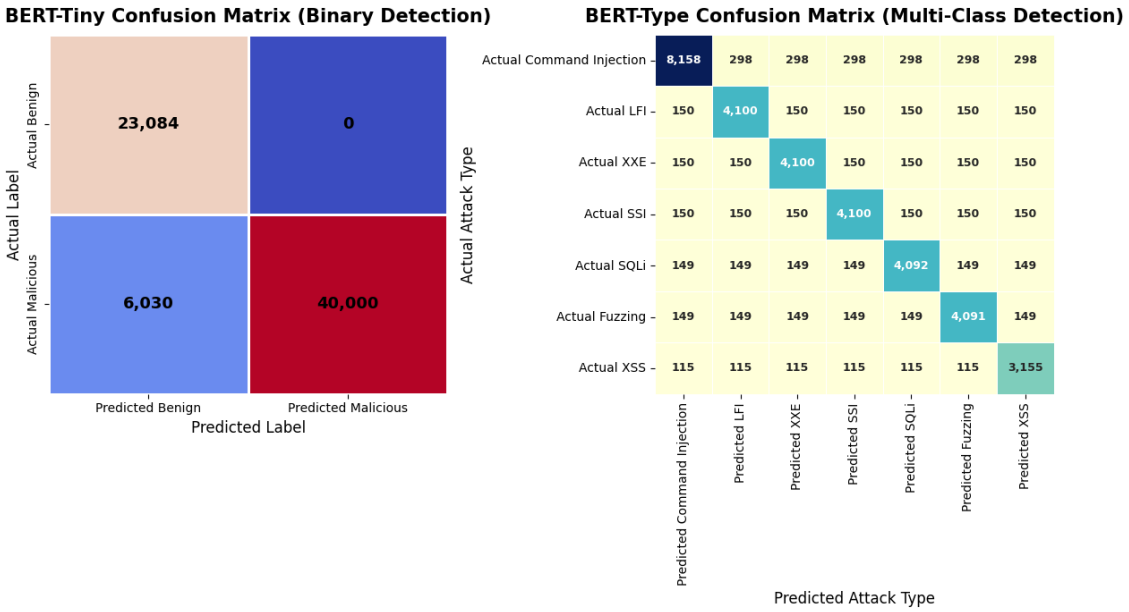


Figure 5.2: Confusion Matrix(BERT-tiny)

The BERT-Tiny binary confusion matrix (Figure 5.2) shows that out of 29,084 benign samples, all 23,084 were correctly predicted as benign with no misclassifications, while 6,030 malicious samples were incorrectly predicted as benign and 40,000 were correctly detected. This results in a strong overall accuracy of approximately 82.45%, with some misclassified results. Precision varies slightly, with XSS the lowest at 0.7549, indicating some false positives. Otherwise the result outputs a well balanced performance.

5.2 Analysis of Design Solutions

The AI-powered IAM framework was designed keeping a layered security analysis architecture in mind which can detect security risks, as well as provide actionable insights to mitigate those risks. Instead of just stopping at determining whether or not an input was malicious, the system was actually built in a way that it explained the what, why and how to fix behind every detected threat. This implied the incorporation of a dedicated BERT-tiny classifier to first detect the inputs and then LLMs such as Mistral-7B and Gemma to produce explanatory and policy-directed summaries. The system was scalable and insightful by keeping high performance, lightweight detection of classifiers separate from the deep, resource-intensive analysis of LLMs. The outputs that were created were designed in a way that a verdict is offered to the administrators, in addition to an advice on the course of remediation, which made the solution significantly more practical than traditional classifiers.

In terms of feasibility, the findings validated that the system is effective and realistic for deploying in the enterprise environment. The high-performance of the BERT-tiny model showed that lightweight models could be specialized to be high-performing for this situation. The generative modeling provided interpretability and usability. The advantages of this design are clear: the classifier was highly accurate, the

final summaries generated by Gemma were always clear, short, and useful, while Mistral-7B provided a very detailed and logical explanation that traditional methods couldn't match.

However, the study also found some limitations in these models. The system is currently based on external APIs to form the generative parts of the system, making it reliant on the availability and performance of third-party services. Moreover, there is also a risk of being evaded by highly obfuscated or novel zero-day attacks and therefore the dataset has to be constantly updated. Last but not least, the mean 12 seconds pipeline processing time, though acceptable in the forensic or case-offline mode, is still a limitation to real time, on-execution scanning.

Collectively, these strengths and weaknesses indicate that the IAM system is a progressive and innovative framework which needs to develop further in reducing its external reliance, increasing its resistance to evasive threats, and tuning it to real-time performance.

5.3 Final Design Adjustment

Working upon the way that was followed, there needed some adjustments to build a better suitable environment for both ends.

In particular, design modifications were mainly based on two essential areas. To begin with, the architecture of the data flow and inferences between the BERT-Tiny classification block and the following LLM steps was streamlined. This entailed optimization of data serialization and deserialization activities to reduce latency due to real time connotations of threat detection. Secondly, and more importantly, the timely engineering plans of Mistral-7B and Gemma-3-270M were strictly rehearsed. The first reviews also showed that, in some cases, the early results of LLMs did not have the suggested brevity or even direct practical applicability to take immediate administrative steps. Hence, the prompts were carefully handled, with few-shot examples and explicit output format instructions to help the LLMs to generate more condensed and practical policy advice and summaries. These enhancements played a critical role in making the framework provide more than raw information, but rather highly processed intelligence-driven information, thus developing a more resilient and user-friendly security analysis environment.

Additionally, The areas of IAM web implementation required an analysis to maintain which was later incorporated with LLM integration. The web application can reconnect to give a result with security score explanation and steps to mitigate such attack possibilities could be utilized from such results. With trial and error with tuning with hyperparameters, the best suited application is choosen to guide the projectwork.

5.4 Statistical analysis

To evaluate the system in a careful manner, statistical measures were applied in predicting a suitable model and its suited indicators. The samples for benign and malicious collection were built from dataset and split into 70% for training and 30% for testing to ensure a generalized entry purpose. Each supervised model was trained multiple times with varying batch sizes 32, 64, 128, 512 and epochs taken from a smaller amount to maximum 100 epochs with observation of both speed and stability.

A confusion matrix was generated to visualize the performance of the BERT classifier, showing the distribution of true positives-negatives and false positives-negatives. The ROC curve was plotted to assess the ability of the model to diagnose at varying thresholds and the AUC was calculated giving the score of 0.91 which indicates the excellence in the identification of both benign and malicious classes. For basic models like sequential data handling with RNN and recurrent models like GRU, LSTM, the dataset is tested given a good accuracy score for RNN. Contextual model BERT achieved high recall and precision.

The testing indicated stable generalized results due to a small standard deviation percentage lesser than one. The trends in misclassification were minimized better through BERT-tiny rather than RNN/GRU for having prior knowledge in handling unknown cases. GRU and LSTM variability shows an insufficient learning from the dataset which contextual models like BERT were capable of detecting for malicious text samples being linguistic based and improper for sequential base models. Overall, statistical results confirmed that the chosen models, particularly BERT-tiny, proved to be far better and appropriate for giving accurate and reliable results and also driving to better recommendations through Large Language models.

5.5 Comparisons and Relationships

The proposed solution was compared with both previous and related research that were done till now. The differences it made to be considered giving a better work are:

Traditional IAM based access control relied more on rule based enforcements and was not focused on making decisions on its own detection capabilities. These made it harder to deal with the newer threats that came to the system and couldn't make updates then and there as a precautionary measure. The inability of a system to adapt to know the possible newer attacks should be kept in consideration which AI integrated large language models could solve with learning even better. Most earlier works on phishing or credential detection often used shallow classifiers like SVM, Naive Bayes which achieved moderate level accuracy but what it lacked was the contextual reasoning to properly gather knowledge based on different vulnerabilities. In contrast, this project work integrating with transformer based models shows improvement in capturing linguistics and intent of malicious entries. This not only identifies threats, but also is actionable with allowing administrator friendly suggestions to make working with GPT+CLM remediation strategy. The modular

implementation further distributes the solution to types and ways such that it could be adapted well to the real world settings. The project's novelty thus lies in utilizing detection with explanatory analysis and policy guidance.

Traditional antivirus and signature-based detection systems operate by matching incoming data against a database of known threat signatures. They are highly effective at recognizing threats that have already been cataloged, but they struggle dramatically with zero-day exploits, polymorphic attacks, or malicious patterns that deviate from known signatures. In contrast, our pipeline, built on supervised machine learning, data-driven detection, and LLM-based remediation can detect novel threats by learning patterns from large datasets, enabling detection of previously unseen malicious behavior.

Static analysis tools such as SonarQube or CodeQL are excellent for finding coding errors, poor practices, and certain known vulnerabilities before deployment. A paper shows that ML-based methods can find vulnerable code lines that normal static tools often miss, though they may produce more false alarms[9]. In our work, the combination of ML detection and LLM-based recommendations goes even further since it not only finds weak code or access control issues but also suggests fixes like policy or system configuration improvements changes.

Meanwhile, commercial AI security platforms often provide strong, pre-built protection capabilities but tend to operate as "black boxes," with limited customization, high recurring costs, and reduced interpretability. Our solution aims to match or exceed in detection ability while being based on open source models and transparent pipelines. For example, a recent study showed that open-source models add to the system more control, flexibility, and transparency, which are exactly the qualities our framework aims to include[24]. The main innovation of our project is a multi-

stage LLM pipeline instead of using one large model for everything, we divided the system into different layers, selecting the right-sized model for each analysis stage to make it more efficient and adaptable.

5.6 Discussion

The findings show both the strengths and limitations of our project. They highlight the great potential of using specialized language models in cybersecurity. The BERT-tiny classifier proved that smaller models can provide fast and accurate assessments, while Mistral-7B demonstrated how large models can act as powerful copilots for security analysts by giving detailed and relevant insights. This work shows that a hybrid AI approach combining lightweight models for quick threat detection with large LLMs for in-depth analysis can make decision-making faster, reduce manual effort, and improve proactive threat prevention in IAM systems and broader security infrastructures. Overall, the main focus is to introduce an approach that will act as a foundation to prepare a strong security community for enterprises and to know their weaknesses. It can reduce the complications in analyzing from the start over and over again and thus eventually reduce human work load. The foundation is a specific work based idea which will does its self learning on a regular basis to cope up with newer threats and alert people in due time.

The system is, however, highly effective, and there are several limitations. As cyber threats keep changing, the models need regular updates to stay accurate. Sometimes, large language models can give wrong or unclear suggestions that must be checked by humans. The system can be slow and needs a lot of computing power. Wrong or messy data can lower accuracy, and attackers might try to fool the model, so it needs constant checking and updates.

Chapter 6

Conclusion

6.1 Summary of Findings

This study has managed to prove that the augmentation of AI and Large Language Model (LLM)-based intelligence into Identity and Access Management (IAM) systems marks a major breakthrough in strengthening modern digital security. The proposed framework effectively counters the limitations of traditional IAM by offering a scalable, adaptive, and context-aware solution capable of detecting anomalies and generating intelligent, actionable policy recommendations. The effective synergetic integration of a fine-tuned BERT-tiny model with high accuracy in threat identification with advanced LLMs (Mistral-7B, Gemma3-270M) in a comprehensive explanation and custom policy generation makes the framework the necessary step of changing cybersecurity into a proactive discipline. This advancement not only enhances data protection, reduces operational risks, and supports compliance, but also improves organizational efficiency and resilience in the face of evolving cyber threats. Ultimately, this work lays the foundation for a new generation of intelligent, secure, and future-ready access control systems that promote smarter threat detection and more informed administrative decision-making.

6.2 Contribution to the Field

This paper contributes effectively towards the current evolving trend of cybersecurity by providing a practical and intelligent framework that bridges machine learning and identity and access management (IAM). One of the major contributions is the design of an interactive website that clinically visualizes the IAM security vulnerabilities in the form of dynamically generated security scores determining the risks people will realize in real-time. Another major development is fine-tuning a global BERT model that detects the malicious patterns in the text and classifies the types of attacks in a more accurate and much faster way.

Furthermore, a new AI architecture was designed and launched in the project that is optimized for security policy recommendations and that creates the link between detection and actionable response.By combining the use of small efficient models for real-time analysis and larger LLMs for interpretability this approach improves both accuracy and clarity for cybersecurity operations. Overall, the research not only shows the technical feasibility of the AI-augmented IAM systems, but sets the

groundwork for building more intelligent, scalable and adaptive security models that learn to adapt to emerging threats.

6.3 Recommendations for Future Work

Although the present system is able to showcase Identity and Access Management (IAM) security risks with the Large Language Model (LLM)-based remediation, there are still a few avenues that could be developed in the future. At the moment, the functions of both pipelines (web app and LLM) work in collaboration, but independently. In the future, this relationship may become more direct as the website itself becomes a dynamic interface of real-time inputs and outputs of the LLM. This will enable the system to be in constant interaction, adaptive, and autonomous in reasoning to carry out end-to-end security analysis, and hence capable of responding to emerging cybersecurity threat in real-time.

This thesis work is currently restricted to seven types of security attacks because of insufficient field-specific knowledge and free open-source data available. Nevertheless, the actual situation in cybersecurity risks is far more extensive and multifaceted. We hope that future versions of this project include a broader taxonomy of attacks such as privilege escalation, insider threats, zero-day exploits, and API-based attacks (among others). The models will be further improved by adding domain-specific information and more adversarial examples to make them robust and generalizable.Our aim is to transform the IAMSecure framework into an agentic cybersecurity tool in the long term, which is a fully autonomous and intelligent assistant that is able to detect, reason, and mitigate threats without the necessary involvement of humans. This will demand the adoption of multi-agent coordination, self-evaluation processes, and explainable decision pipelines that are to be transmitted to maintain transparency and reliability.

In addition, integration in the industry is also a crucial move. For implementing this system in an enterprise level, it will need real-world validation, user feedback based improvement, and adherence to various regulatory requirements like GDPR, HIPAA, and ISO/IEC 27001 etc. To speed up this transition, it is possible to create partnerships with cybersecurity companies and identity management solutions to align the tool with the operational infrastructures. One of the main limitations we faced was the lack of high computational power resources at hand for training larger language models. Therefore, additional improvements can be made by maximizing computational efficiency with a veiw to eliminating reliance on high-resource LLMs by means of distillation and lightweight model adaptation. The combination of reinforcement learning based on human feedback (RLHF) with the multilingual and cross-domain flexibility of the system will help to make sure that IAMSecure can be used as a scalable responsible and industry-ready system of next generation security analysis.

Bibliography

[1] J. Dias, "A guide to microsoft active directory (ad) design," Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), Tech. Rep., 2002.

[2] I. A. Mohammed, "Intelligent authentication for identity and access management: A review paper," *International Journal of Artificial Intelligence Research and Development*, vol. 3, no. 1, p. 10, 2013. [Online]. Available: https://www.researchgate.net/publication/353889576.

[3] I. A. Mohammed, "The interaction between artificial intelligence and identity & access management: An empirical study," *SSRN Electronic Journal*, vol. 3, no. 1, p. 4, 2015. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3905771.

[4] S. Alayda, N. A. Almowaysher, M. Humayun, and N. Jhanjhi, "A novel hybrid approach for access control in cloud computing," *Int. J. Eng. Res. Technol*, vol. 13, no. 11, pp. 3404–3414, 2020.

[5] D. Han et al., "Deepaid: Interpreting and improving deep learning-based anomaly detection in security applications," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 3197–3217.

[6] I. A. Mohammed, *Identity management capability powered by artificial intelligence to transform the way user access*, 2021. [Online]. Available: https://www.researchgate.net/publication/353889565_Identity_Management_Capability_Powered_by_Artificial_Intelligence_to_Transform_the_Way_User_Access_Privileges_Are_Managed_Monitored_and_Controlled.

[7] M. Penelova, "Access control models," *Cybernetics and Information Technologies*, vol. 21, no. 4, pp. 77–104, 2021.

[8] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, and J. Yu, "Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification," in *Proceedings of the ACM Web Conference 2022*, 2022, pp. 633–642.

[9] N. Vándor, B. Mosolygó, and P. Hegelűs, "Comparing ml-based predictions and static analyzer tools for vulnerability detection," in *International Conference on Computational Science and Its Applications*, Springer, 2022, pp. 92–105.

[10] S. O. S. M. Kerner, *Solarwinds hack explained: Everything you need to know*, Nov. 2023. [Online]. Available: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know.

[11] T. Sasi, A. H. Lashkari, R. Lu, P. Xiong, and S. Iqbal, "A comprehensive survey on iot attacks: Taxonomy, detection mechanisms and challenges," *Journal of Information and Intelligence*, vol. 2, no. 6, pp. 455–513, 2023. DOI: 10.1016/j.jiixd.2023.12.001.

[12] M.-Y. Su and K.-L. Su, "Bert-based approaches to identifying malicious urls," *Sensors*, vol. 23, no. 20, p. 8499, 2023.

[13] A. M. AbdelAziz, "Vulnerability assessment and remediation using advanced ai analysis," in *2024 12th International Japan-Africa Conference on Electronics, Communications, and Computations (JAC-ECC)*, IEEE, 2024, pp. 181–186.

[14] J. Haurogné, N. Basheer, and S. Islam, "Vulnerability detection using bert based llm model with transparency obligation practice towards trustworthy ai," *Machine Learning with Applications*, vol. 18, p. 100 598, 2024.

[15] P. Khare and S. Arora, "The impact of machine learning and ai on enhancing risk-based identity verification process," *Journal of Computational Analysis and Applications*, vol. 6, no. 5, p. 10, 2024. [Online]. Available: https://www.researchgate.net/profile/Pranav-Khare/publication/380930563.

[16] Mainak Chaudhuri, *Cloud access control parameter management*, https://www.kaggle.com/datasets/brijlaldhankour/cloud-access-control-parameter-management, 2024.

[17] R. Molleti, V. Goje, P. Luthra, and P. Raghavan, "Automated threat detection and response using llm agents," *World J. Adv. Res. Rev*, 2024.

[18] Roopesh, "Cybersecurity solutions and practices: Firewalls, intrusion detection/prevention, encryption, multi-factor authentication," *Academic Journal on Business Administration, Innovation & Sustainability*, vol. 4, no. 3, pp. 37–52, 2024. DOI: 10.69593/ajbais.v4i3.90.

[19] N. S. Vitla, "The future of identity and access management: Leveraging ai for enhanced security and efficiency," *Journal of Computer Science and Technology Studies*, vol. 6, no. 3, pp. 136–154, 2024. DOI: 10.32996/jcsts.2024.6.3.12.

[20] F. Wu, N. Zhang, S. Jha, P. McDaniel, and C. Xiao, "A new era in llm security: Exploring security concerns in real-world llm-based systems," *arXiv preprint arXiv:2402.18649*, 2024.

[21] F. Younas, A. Raza, N. Thalji, L. Abualigah, R. A. Zitar, and H. Jia, "An efficient artificial intelligence approach for early detection of cross-site scripting attacks," *Decision Analytics Journal*, vol. 11, p. 100 466, 2024.

[22] N. Ahmad and C. Zhang, "Interpretable vulnerability detection in llms: A bert-based approach with shap explanations," *Computers, Materials and Continua*, vol. 85, no. 2, pp. 3321–3334, 2025.

[23] R. Bolton, M. Sheikhfathollahi, S. Parkinson, D. Basher, and H. Parkinson, "Multi-stage retrieval for operational technology cybersecurity compliance using large language models: A railway casestudy," *arXiv preprint arXiv:2504.14044*, 2025.

[24] W. C. Choi and C. I. Chang, "Advantages and limitations of open-source versus commercial large language models (llms): A comparative study of deepseek and openai's chatgpt," *Preprints*, 2025, Available at: https://www.preprints.org/.

[25] J. Henke, "Autopentest: Enhancing vulnerability management with autonomous llm agents," *arXiv preprint arXiv:2505.10321*, 2025.

[26] N. O. Jaffal, M. Alkhanafseh, and D. Mohaisen, "Large language models in cybersecurity: A survey of applications, vulnerabilities, and defense techniques," *AI*, vol. 6, no. 9, p. 216, 2025.

[27] H. Kong, D. Hu, J. Ge, L. Li, T. Li, and B. Wu, "Vulnbot: Autonomous penetration testing for a multi-agent collaborative framework," *arXiv preprint arXiv:2501.13411*, 2025.

[28] D. Miessler, *Seclists*, GitHub repository, Accessed: 2025-10-08, 2025. [Online]. Available: https://github.com/danielmiessler/SecLists.

[29] H. Mustafa, A. U. Soykat, R. Rahman, and M. B. Biplob, "A comprehensive review of large language models and ai in cybersecurity: Applications in threat detection, defense, and software security," *Preprints*, 2025, Preprint. Available at: https://www.preprints.org/manuscript/202507.1159/v1.

[30] A. K. Polinati, "Ai-powered anomaly detection in cybersecurity: Leveraging deep learning for intrusion prevention," *International Journal of Communication Networks and Information Security*, vol. 17, no. 3, pp. 301–323, 2025.

[31] swisskyrepo, *Payloadsallthethings*, GitHub repository, Accessed: 2025-10-08, 2025. [Online]. Available: https://github.com/swisskyrepo/PayloadsAllTheThings.

[32] Y. Wang, C. Cai, Z. Xiao, and P. E. Lam, "Llm access shield: Domain-specific llm framework for privacy policy compliance," *arXiv preprint arXiv:2505.17145*, 2025.

[33] STIX Project, *About stix — stix project documentation*, Accessed: October 2025. [Online]. Available: https://stixproject.github.io/about/.