INTERNATIONAL JOURNAL OF PHARMACY WITH MEDICAL SCIENCES
Peer Reviewed, Referred & Indexed Journal
E-ISSN:3068-2738
Vol.6, No. 2(1)(2026)
www.ijpams.com

AI-Powered Regulatory Compliance Checker for Contracts

Mr. Bhaba Sankar Patra
Student, Dept. of CSE-AI,
GIFT Autonomous, Bhubaneswar

Mr. Gulsan Ram
Student, Dept. of CSE-AI,
GIFT Autonomous, Bhubaneswar

Dr. Soumendra Prasad Rout
Assistant Professor, Dept. of CSE-AI,
GIFT Autonomous, Bhubaneswar

***Abstract*— The rapid growth of digital contracts across industries such as healthcare, finance, information technology, and government sectors has increased the complexity of regulatory compliance management. Traditional manual compliance verification methods are time-consuming, expensive, and prone to human errors. This research paper presents an AIpowered Regulatory Compliance Checker for Contracts that leverages Natural Language Processing (NLP), Machine Learning (ML), and Large Language Models (LLMs) to automate the analysis of legal agreements and identify compliance risks. The proposed system extracts contractual clauses, maps them with regulatory frameworks such as GDPR, HIPAA, PCI-DSS, and organizational policies, and detects missing or non-compliant clauses. The architecture integrates document ingestion, clause classification, semantic analysis, retrieval-augmented generation (RAG), and risk scoring mechanisms. Experimental evaluation demonstrates improved efficiency, accuracy, scalability, and consistency in contract analysis compared to traditional manual review systems. The research highlights the potential of AI-driven legal technology to transform regulatory compliance management while reducing operational costs and legal risks.**

Keywords*— *Artificial Intelligence, Regulatory Compliance, Contracts, NLP, LLM, Machine Learning, LegalTech, Risk Analysis, Clause Extraction

I. INTRODUCTION

In the modern digital era, organizations generate and manage a massive number of contracts for business operations, partnerships, employment, procurement, healthcare services, financial transactions, and data-sharing agreements. These contracts contain important legal clauses, regulatory obligations, confidentiality agreements, payment conditions, liability terms, and compliance requirements. Ensuring that every contract follows applicable laws and industry regulations is a critical responsibility for organizations. However, manual contract review is a complex, timeconsuming, and error-prone process that requires significant legal expertise and operational resources.

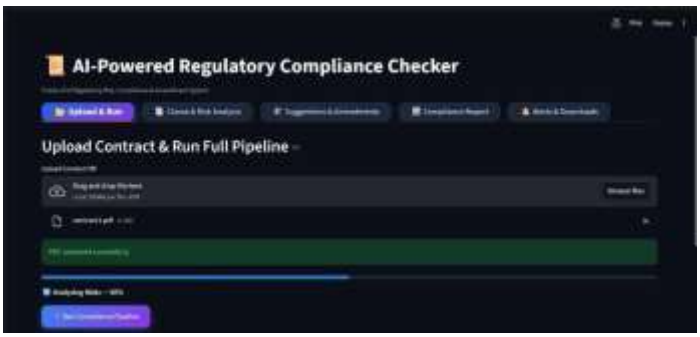
With the rapid growth of regulations such as the General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), Payment Card Industry Data Security Standard (PCI-DSS), Sarbanes-Oxley Act (SOX), and ISO compliance standards, organizations face increasing challenges in maintaining regulatory compliance across all contractual documents. Failure to comply with these regulations may result in severe financial penalties, legal disputes, reputational damage, and operational risks. Traditional rule-based compliance systems are often limited because they rely heavily on keyword matching and cannot fully understand the semantic meaning and contextual interpretation of legal language.

Recent advancements in Artificial Intelligence (AI), intelligent filtering techniques, and dashboard-based cybersecurity platforms have improved the efficiency of vulnerability analysis and visualization [4]. Modern dashboard systems enable centralized monitoring of vulnerabilities through interactive visualizations, severity analysis, and workflow management. AI-assisted filtering and contextual search mechanisms further enhance vulnerability prioritization and reduce manual analysis efforts [5]. However, many existing systems still lack integrated task management, intelligent filtering, scalable role-based governance, and centralized workflow coordination.

To address these limitations, this paper proposes an AI Vulnerability Management Dashboard, a centralized webbased cybersecurity platform designed to improve vulnerability analysis, prioritization, and remediation workflows. The system is developed using modern full-stack technologies including React, Tailwind CSS, Flask, SQLAlchemy, and PostgreSQL to ensure modularity, scalability, and secure API communication [6].

The dashboard provides real-time visibility into vulnerabilities through interactive graphs, severity distribution analysis, and centralized management interfaces. The system also incorporates essential security mechanisms such as RoleBased Access Control (RBAC), Content Security Policy (CSP), and Cross-Site Request Forgery (CSRF) protection to ensure secure operation [7].

Figure-1: AI Powered Regulatory Compliance Checker



In addition to standard user and administrator functionalities, the system architecture conceptually supports a hierarchical Super Admin module for enterprise-level governance and centralized monitoring. The inclusion of hierarchical Artificial Intelligence (AI) has emerged as a transformative technology capable of automating complex document analysis tasks. Recent advancements in Natural Language Processing (NLP), Machine Learning (ML), and Large Language Models (LLMs) have significantly improved the ability of computers to understand, interpret, and analyze human language. These technologies provide an opportunity to automate legal contract analysis and compliance verification with higher speed, consistency, and accuracy.

EXISTING APPROACHES

The rapid increase in cybersecurity threats and the continuous evolution of digital infrastructures have significantly increased the importance of vulnerability management systems in modern organizations. Vulnerability management involves identifying, analyzing, prioritizing, and mitigating security weaknesses within software systems, web applications, APIs, and network infrastructures. Various approaches and technologies have been developed to improve vulnerability detection and risk management processes [8].

Traditional vulnerability management systems such as Nessus, OpenVAS, Qualys, and Rapid7 Nexpose are widely used for automated vulnerability scanning and assessment. These tools primarily identify vulnerabilities by comparing system configurations and software behaviors against known vulnerability databases such as Common Vulnerabilities and Exposures (CVE) [9]. Most of these systems use the Common Vulnerability Scoring System (CVSS) to classify vulnerabilities based on severity levels and provide remediation recommendations.

Although traditional vulnerability scanning systems are highly effective in detecting vulnerabilities, they often generate large amounts of security data that require manual interpretation and prioritization. Security analysts are responsible for reviewing extensive vulnerability reports, analyzing contextual relevance, and determining remediation priorities manually. In enterprise environments, where thousands of vulnerabilities may exist simultaneously, this process becomes highly time-consuming and operationally inefficient [10].

The Common Vulnerability Scoring System (CVSS) has become one of the most widely adopted standards for vulnerability severity assessment. CVSS calculates vulnerability scores based on multiple parameters such as attack vector, attack complexity, privileges required, user interaction, confidentiality impact, integrity impact, and availability impact [11]. This standardized approach enables organizations to classify vulnerabilities into categories such as Low, Medium, High, and Critical.

Despite its widespread adoption, CVSS-based assessment mechanisms have certain limitations. CVSS primarily provides static severity evaluation and does not always consider contextual or environmental factors such as organizational priorities, asset criticality, exploit trends, or operational dependencies. As a result, vulnerabilities with similar CVSS scores may have different levels of impact in different environments [12]. This limitation highlights the need for intelligent filtering and contextual prioritization mechanisms within modern vulnerability management systems.

Recent advancements in Artificial Intelligence (AI) and intelligent cybersecurity systems have introduced new approaches for improving vulnerability analysis and threat prioritization. AI-assisted cybersecurity systems are capable of analyzing large datasets, identifying patterns, detecting anomalies, and automating decision-making processes [13].

Machine learning algorithms have been widely applied in areas such as malware detection, intrusion detection systems, phishing analysis, behavioral monitoring, and vulnerability prediction.

AI-driven vulnerability management systems can improve prioritization accuracy by analyzing historical vulnerability patterns and identifying high-risk security issues automatically. However, advanced machine learning systems often require large training datasets, high computational resources, and continuous model optimization. These factors increase implementation complexity and maintenance overhead in practical cybersecurity environments [14].

Table-I: Comparison of Traditional System and Proposed system

Feature	Traditional Tools	Proposed System
Vulnerability Detection	Strong	Strong
Intelligent Filtering	Limited	Integrated
Dashboard Visualization	Basic	Advanced
Task Management	Limited	Integrated
Role-Based Access	Basic	Advanced
Super Admin Support	No	Supported

To balance efficiency and implementation feasibility, many modern cybersecurity systems implement lightweight rulebased intelligent filtering approaches that provide AI-assisted functionality without requiring full-scale machine learning infrastructure. Intelligent filtering mechanisms can improve search efficiency, contextual vulnerability analysis, and severity prioritization while maintaining lightweight system performance and scalability [15].

Dashboard-based cybersecurity platforms have also become increasingly important in modern security operations. Security dashboards provide centralized visualization of vulnerability data through interactive graphs, severity indicators, charts, and analytics panels. These systems improve usability by allowing security analysts to monitor vulnerabilities, track remediation

workflows, and analyze security trends from a single interface [16].

Modern dashboard systems commonly include functionalities such as:

- Vulnerability tracking
- Severity visualization
- Search and filtering
- Task management
- User activity monitoring
- Real-time analytics

Despite these advancements, many existing dashboard platforms mainly focus on visualization and reporting rather than intelligent analysis and workflow integration. Most systems provide static filtering capabilities and limited contextual search functionality. Additionally, many existing systems do not support hierarchical administrative structures required for enterprise-level governance and centralized monitoring.

Modern vulnerability management platforms also increasingly rely on secure web-based architectures and RESTful APIs for communication between frontend and backend systems. Technologies such as React, Flask, Node.js, SQLAlchemy, and PostgreSQL are widely used for developing scalable cybersecurity applications [17]. Security mechanisms such as Role-Based Access Control (RBAC), Content Security Policy (CSP), secure authentication, and Cross-Site Request Forgery (CSRF) protection are essential for protecting web-based cybersecurity platforms against unauthorized access and malicious attacks [18].

The analysis of existing approaches reveals several important limitations in current vulnerability management systems:

a. Over-reliance on manual vulnerability prioritization
b. Lack of intelligent contextual filtering
c. Limited integration of remediation workflows
d. Poor centralized visualization in traditional systems
e. High complexity of advanced AI-based platforms
f. Limited hierarchical administrative control

These limitations highlight the need for a centralized, scalable, and intelligent vulnerability management platform capable of integrating CVSS-based assessment, intelligent filtering, dashboard visualization, secure role-based access control, and workflow-oriented task management into a single unified system.

The proposed AI Vulnerability Management Dashboard addresses these challenges by combining structured vulnerability analysis, intelligent filtering, secure API architecture, interactive dashboard visualization, and scalable administrative control mechanisms within a modern full-stack cybersecurity platform.

II. PROPOSED SYSTEM ARCHITECTURE

The proposed AI Vulnerability Management Dashboard is designed as a centralized and scalable cybersecurity platform that integrates vulnerability analysis, intelligent filtering, task management, and secure access control into a unified system. The architecture follows a modular full-stack design approach to ensure scalability, maintainability, secure communication, and efficient processing of vulnerability data in modern cybersecurity environments [19].

The system is developed using modern web technologies including React, Tailwind CSS, Flask, SQLAlchemy, and PostgreSQL. The frontend layer provides an interactive and responsive user interface for monitoring vulnerabilities and managing workflows, while the backend layer handles business logic, API processing, intelligent filtering, CVSS computation, and role-based access control. The database layer securely stores vulnerability records, user information, task data, and CVSS-related metrics [20].

The proposed architecture follows a multi-layered clientserver model consisting of the following major components:

- User Interface Layer
- Frontend Application Layer
- Backend API Layer
- ORM Layer
- Database Layer
- Security Layer

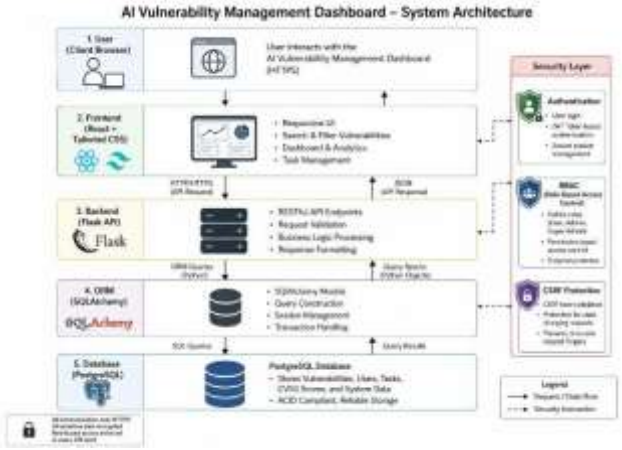
The User Interface layer acts as the entry point of the system where users interact with the dashboard through a web browser. Users can search vulnerabilities, monitor severity distribution, assign remediation tasks, and manage operational workflows through interactive dashboard interfaces. The frontend layer is implemented using React and Tailwind CSS to provide responsive layouts, dynamic component rendering, and efficient user interaction [21].

The backend layer is implemented using Flask REST APIs and acts as the core processing unit of the system. The backend handles request validation, authentication, business logic execution, CVSS score computation, intelligent filtering operations, and response generation. RESTful APIs enable secure communication between frontend and backend modules using JSON-based request-response mechanisms [22].

SQLAlchemy ORM is used to manage database communication and structured data handling. The ORM layer simplifies query construction, session management, and transaction handling while improving database maintainability and scalability. PostgreSQL is used as the primary database system for securely storing vulnerabilities, users, task assignments, CVSS scores, and operational records [23].

Figure-2: Overall System Architecture of AI Vulnerability Management Dashboard

The architecture also integrates multiple security mechanisms to ensure secure system operation and protection against common web vulnerabilities. Role-Based Access Control (RBAC) is implemented to restrict system access based on user roles such as User, Admin, and Super Admin. Content Security Policy (CSP) and Cross-Site Request Forgery (CSRF) protection mechanisms are integrated to prevent unauthorized requests, malicious scripts, and session manipulation attacks [24].



A significant feature of the proposed architecture is the conceptual implementation of a **Super Admin Command Center** for enterprise-level governance and centralized monitoring. The Super Admin module enables global oversight of organizations, users, vulnerabilities, and remediation workflows through advanced dashboard analytics and monitoring interfaces. This hierarchical administrative structure improves scalability and supports centralized governance across multiple operational environments.

The overall workflow of the proposed system begins when vulnerability data is submitted through the user interface or APIs. The backend validates and preprocesses incoming data before applying CVSS score computation and intelligent filtering mechanisms. The processed data is stored securely in the database and visualized through the dashboard interface using graphs, severity indicators, and analytical charts. The system also supports remediation task assignment and tracking functionalities to improve operational coordination [25].

The modular architecture of the system provides several advantages including:

- Improved scalability and maintainability
- Secure API communication
- Centralized vulnerability monitoring
- Efficient vulnerability prioritization
- Interactive dashboard visualization
- Structured workflow management
- Enterprise-level administrative control

The integration of intelligent filtering, CVSS scoring, secure APIs, and centralized dashboard visualization enables the proposed architecture to provide a comprehensive vulnerability management solution compared to traditional vulnerability management systems.

Table-II: Comparison of Traditional System and Proposed system

III. METHODOLOGY

The development of the AI Vulnerability Management Dashboard follows a structured and modular methodology to ensure efficient vulnerability processing, secure communication, intelligent prioritization, and scalable system implementation. The methodology mainly focuses on vulnerability data handling, backend processing, CVSS-based

Layer	Technology Used
Frontend	React, Tailwind CSS
Backend	Flask REST API
ORM	SQLAlchemy
Database	PostgreSQL
Security	RBAC, CSP, CSRF
Dashboard	React Charts & Analytics

severity assessment, intelligent filtering, dashboard visualization, and remediation workflow management [26].

The proposed system follows a workflow-oriented architecture where vulnerability data is processed through multiple stages before being visualized on the dashboard. The overall methodology begins with vulnerability data input through user interfaces or APIs. The backend validates and preprocesses incoming data to remove inconsistencies and ensure structured formatting before applying CVSS-based severity computation [27].

The system workflow consists of the following major stages:

a. Data Input
b. Backend Validation and Preprocessing
c. CVSS Score Computation
d. Database Storage
e. Intelligent Filtering and Querying
f. Dashboard Visualization
g. Task Assignment and Tracking

Initially, vulnerability data is submitted by users or external systems through frontend interfaces or APIs. The backend layer validates incoming requests, checks data consistency, and preprocesses vulnerability information before storing it in the database. This preprocessing stage improves data integrity and ensures standardized vulnerability handling throughout the system.

Figure-3: Overall Workflow of AI Vulnerability Management Dashboard

The Common Vulnerability Scoring System (CVSS) is integrated into the methodology to provide standardized vulnerability severity assessment. CVSS scores are calculated based on multiple parameters such as attack vector, attack complexity, privileges required, confidentiality impact, integrity impact, and availability impact [28]. The computed scores are then categorized into severity levels including Low,

Medium, High, and Critical to support vulnerability prioritization and remediation planning.

To improve search efficiency and contextual analysis, the system implements a lightweight intelligent filtering mechanism. Instead of using computationally expensive machine learning models, the proposed system adopts rulebased intelligent filtering techniques that provide AI-assisted functionality while maintaining lightweight performance and scalability [29]. The filtering mechanism allows users to search vulnerabilities dynamically based on titles, severity levels, descriptions, keywords, and contextual attributes.

The backend system is implemented using Flask REST APIs, which handle authentication, request processing, business logic execution, database communication, and response generation. RESTful APIs provide secure and structured communication between frontend and backend modules using JSON-based request-response mechanisms [30]. SQLAlchemy ORM is used to manage database operations, query construction, and transaction handling efficiently.

The methodology also incorporates secure system communication and access control mechanisms to ensure safe operation of the platform. Role-Based Access Control (RBAC) is implemented to restrict system functionalities based on user roles such as User, Admin, and Super Admin. Additional security mechanisms including Content Security Policy (CSP), secure authentication, and Cross-Site Request Forgery (CSRF) protection are integrated to prevent unauthorized access and malicious requests [31].

Task management and remediation tracking are integrated into the workflow to improve operational coordination among users. Once vulnerabilities are identified and prioritized, administrators can assign remediation tasks to users directly through the dashboard. The task management module enables tracking of remediation progress through different states such as Open, In Progress, and Resolved. This centralized workflow improves vulnerability resolution efficiency and reduces manual coordination overhead [32].

The proposed methodology also supports scalability and future extensibility. The modular system design enables integration of additional functionalities such as automated vulnerability scanners, cloud deployment, machine learningbased predictive analysis, and real-time threat intelligence feeds in future versions of the platform.

The implementation methodology provides several advantages:

- Standardized vulnerability severity assessment
- Improved contextual vulnerability filtering
- Secure API-based communication
- Centralized workflow management
- Interactive dashboard visualization
- Efficient remediation tracking

The structured methodology adopted in this system improves vulnerability analysis efficiency, reduces manual prioritization efforts, and enhances operational visibility compared to traditional vulnerability management approaches.

IV. SYSTEM DESIGN AND IMPLEMENTATION

The AI Vulnerability Management Dashboard was successfully implemented as a full-stack web-based cybersecurity platform integrating frontend visualization, backend processing, intelligent filtering, vulnerability scoring, and centralized workflow management. The implementation focuses on scalability, secure communication, modular architecture, and efficient vulnerability handling to support modern cybersecurity operations [33].

The frontend of the system is developed using **React.js** and **Tailwind CSS** to provide a responsive and interactive user interface. React's component-based architecture enables modular UI development and efficient rendering of dashboard components such as vulnerability lists, severity graphs, search modules, and task management panels [34]. Tailwind CSS is used to improve responsiveness, alignment consistency, and modern dashboard visualization across different devices and screen resolutions.

The backend system is implemented using **Flask REST APIs**, which handle request processing, business logic execution, CVSS score computation, intelligent filtering, authentication, and database communication. Flask was selected due to its lightweight architecture, flexibility, and efficient API integration capabilities [35]. RESTful APIs enable secure JSON-based communication between frontend and backend modules, ensuring smooth interaction and real-time dashboard updates.

The database layer is implemented using **PostgreSQL** with **SQLAlchemy ORM** for structured data storage and management. The database stores vulnerabilities, users, task assignments, severity scores, remediation workflows, and operational records securely [36]. SQLAlchemy simplifies query construction, transaction handling, and database scalability while improving maintainability of backend operations.

Figure-4: Database Schema from DB Browser (SQLite)

The implementation also integrates multiple security mechanisms to ensure secure system operation. Role-Based Access Control (RBAC) is used to restrict functionalities based on user roles such as User, Admin, and Super Admin. Additional security measures including Content Security Policy (CSP), secure authentication, and Cross-Site Request Forgery (CSRF) protection are implemented to prevent unauthorized access and malicious requests [37].

One of the major functionalities implemented in the system is the **CVSS-based vulnerability scoring module**. The backend processes vulnerability parameters and computes severity scores based on CVSS metrics such as attack vector, attack complexity, privileges required, and impact factors. The computed scores are categorized into severity levels including Low, Medium, High, and Critical to improve vulnerability prioritization and remediation planning [38].

The intelligent filtering module was implemented using rulebased contextual filtering techniques. Users can dynamically search vulnerabilities based on keywords, descriptions, severity levels, and vulnerability categories. The filtering system improves vulnerability analysis efficiency and reduces manual search complexity while maintaining lightweight performance [39].

The dashboard visualization module provides centralized monitoring and analytical insights through interactive charts, graphs, severity indicators, and operational statistics. The dashboard displays:

- Total vulnerabilities
- Severity distribution
- Active tasks
- Vulnerability trends
- Team workload analysis
- Global operational statistics
- Recent activity feeds

These visual analytics improve vulnerability visibility and support faster decision-making by security analysts and administrators.

The experimental evaluation of the system focused on functionality testing, dashboard responsiveness, API performance, filtering efficiency, and security validation. During implementation testing, the system successfully processed vulnerability data, computed CVSS scores, and updated dashboard analytics dynamically without significant delay.

The frontend interface remained responsive during continuous vulnerability updates and supported efficient rendering of charts and severity distributions. API response times remained stable during multiple request-response operations, while PostgreSQL provided efficient storage and retrieval of vulnerability records and operational data.

The intelligent filtering mechanism successfully returned contextual vulnerability results based on user search inputs. Dashboard analytics also provided clear visibility into vulnerability trends and remediation workflows. Security testing validated the effectiveness of RBAC, CSP, and CSRF protection mechanisms in preventing unauthorized access and malicious operations.

The Super Admin dashboard implementation further improved centralized governance and operational monitoring. The dashboard provided visibility into organization-wide vulnerabilities, active users, remediation activities, and security trends through advanced analytics panels and centralized administrative controls.

The implementation results demonstrate that the proposed system significantly improves vulnerability prioritization, operational visibility, workflow coordination, and remediation management compared to traditional vulnerability management systems. The integration of intelligent filtering, centralized dashboard visualization, secure APIs, and modular architecture provides a scalable and efficient cybersecurity solution for modern enterprise environments.

V. RESULTS AND DISCUSSION

The implementation and experimental evaluation of the proposed AI Vulnerability Management Dashboard demonstrate significant improvements in vulnerability analysis, prioritization, and workflow management compared to traditional vulnerability management approaches. The integration of intelligent filtering, dashboard visualization, CVSS-based scoring, and centralized remediation tracking provides a more efficient and scalable cybersecurity management platform for modern enterprise environments
[40].

One of the major advantages observed during implementation is the improvement in vulnerability prioritization and operational visibility. Traditional vulnerability management systems often generate extensive reports that require manual interpretation and analysis by security teams. In contrast, the proposed dashboard centralizes vulnerability information through interactive visualization and severity-based categorization, enabling security analysts to identify critical vulnerabilities more efficiently [41].

The integration of CVSS scoring mechanisms significantly improved vulnerability assessment consistency and prioritization accuracy. Vulnerabilities were successfully

categorized into Low, Medium, High, and Critical severity levels based on standardized CVSS parameters. This structured severity classification helped administrators prioritize remediation workflows and allocate resources more effectively.

The intelligent filtering mechanism implemented in the system also improved vulnerability analysis efficiency. Users were able to dynamically search vulnerabilities based on severity levels, contextual keywords, descriptions, and vulnerability categories. Compared to traditional static filtering systems, the proposed filtering approach reduced manual search complexity and improved the speed of vulnerability identification and analysis [42].

Dashboard visualization played a critical role in improving operational monitoring and decision-making. Interactive charts, severity indicators, analytics panels, and remediation tracking modules provided centralized visibility into vulnerability trends, active tasks, and organizational risk levels. Security analysts and administrators could monitor operational status and remediation progress from a single interface, improving coordination and reducing workflow fragmentation.

The integration of Role-Based Access Control (RBAC), secure authentication, Content Security Policy (CSP), and Cross-Site Request Forgery (CSRF) protection mechanisms enhanced the overall security of the platform. Security validation demonstrated that the implemented protection mechanisms successfully prevented unauthorized access attempts and malicious request execution [43]. The hierarchical administrative structure also improved governance by separating operational privileges between Users, Admins, and Super Admins.

The implementation of the conceptual **Super Admin Command Center** further strengthened centralized governance and scalability. The Super Admin module enabled organization-wide monitoring of vulnerabilities, users, remediation activities, and operational statistics through centralized dashboard analytics. This hierarchical governance model improves scalability and makes the proposed system more suitable for enterprise-level deployments and multiorganizational cybersecurity management environments.

Despite the advantages and successful implementation results, several limitations were identified during system development and testing. The current system primarily relies on structured or manually entered vulnerability data and does not include integration with automated vulnerability scanning engines such as Nmap, Nessus, or OWASP ZAP. As a result, vulnerability identification still depends on external scanning systems [44].

Additionally, the intelligent filtering mechanism implemented in the system is rule-based rather than fully machine learningdriven. Although the lightweight filtering approach improves contextual analysis and maintains scalability, advanced AIbased predictive analysis and anomaly detection functionalities are not currently implemented. Integration of machine learning models may further improve automated prioritization and threat prediction capabilities in future versions of the system.

Another limitation involves the absence of real-time threat intelligence integration. The current implementation does not connect with live CVE databases, threat feeds, or external security intelligence services. Real-time threat synchronization and vulnerability feed integration would significantly improve operational awareness and automated vulnerability tracking capabilities.

Future enhancements of the proposed system may include:

- Integration with automated vulnerability scanning tools
- Real-time threat intelligence synchronization
- Machine learning-based vulnerability prediction
- Cloud-native deployment and distributed architecture
- DevSecOps pipeline integration
- Multi-tenant enterprise support
- Advanced reporting and analytics modules

Cloud deployment and DevSecOps integration will further improve scalability, automation, and operational efficiency. Integration with CI/CD security pipelines can enable continuous vulnerability monitoring during software development and deployment processes. Advanced analytics and AI-based prediction models can also improve proactive threat management capabilities in enterprise environments.

The overall implementation and evaluation results demonstrate that the proposed AI Vulnerability Management Dashboard successfully improves vulnerability visibility, prioritization, remediation coordination, and centralized governance compared to traditional vulnerability management systems. The combination of intelligent filtering, secure APIs, dashboard visualization, and modular architecture provides a scalable foundation for future enterprise-level cybersecurity operations.

VI. FUTURE ENHANCEMENTS

The proposed AI Vulnerability Management Dashboard provides a scalable and efficient foundation for modern vulnerability management and cybersecurity operations. Although the current implementation successfully integrates vulnerability tracking, CVSS-based severity assessment, intelligent filtering, and workflow management, several advanced enhancements can further improve system, automation, scalability, and enterprise applicability [45].

One of the major future enhancements involves the integration of automated vulnerability scanning tools such as **Nmap**, **OWASP ZAP**, **Nessus**, and **OpenVAS**. The current implementation primarily relies on structured or manually submitted vulnerability data. This enhancement would significantly reduce manual effort and improve vulnerability identification efficiency in enterprise environments [46].

Future versions of the system can also integrate **real-time threat intelligence feeds** and live vulnerability databases such

as CVE repositories and security advisory APIs. Real-time synchronization with external threat intelligence sources would improve operational awareness and enable automatic updating of newly discovered vulnerabilities, exploit trends, and risk indicators [48]. This integration would enhance the system's ability to support continuous vulnerability monitoring and dynamic threat assessment.

Enhancement	Future Benefit
AI Integration	Predictive vulnerability analysis
Scanner Integration	Real-time vulnerability detection
Cloud Deployment	Improved scalability
DevSecOps Integration	Continuous security validation
Super Admin Expansion	Centralized enterprise governance

Table-III: Future Enhancement Opportunities

Cloud-native deployment and distributed architecture represent another major area for future enhancement. Deploying the platform on cloud environments such as **Amazon Web Services (AWS)**, **Microsoft Azure**, or **Google Cloud Platform (GCP)** would improve scalability, high availability, and distributed data management capabilities [49].

Future implementation can also extend the conceptual **Super Admin Command Center** into a fully operational enterprise governance module. The advanced Super Admin system can provide:

a. Multi-tenant organization management
b. Global vulnerability monitoring
c. User activity auditing
d. System-wide policy configuration
e. Centralized compliance management
f. Cross-organization analytics

Another important enhancement area involves **DevSecOps integration** and continuous security automation. Integration with CI/CD pipelines such as Jenkins, GitHub Actions, GitLab CI/CD, and Docker-based deployment environments would enable automated vulnerability analysis during software development and deployment stages [50]. Continuous security integration would improve proactive vulnerability detection and strengthen secure software development practices.

The modular architecture of the proposed AI Vulnerability Management Dashboard provides strong flexibility for integrating these future enhancements without requiring significant architectural redesign. The scalability and extensibility of the system ensure that it can evolve into a comprehensive enterprise-grade cybersecurity management platform capable of addressing emerging threats and evolving security requirements.

The future enhancement possibilities demonstrate the longterm applicability and adaptability of the proposed system in modern cybersecurity environments. With integration of automation, cloud technologies, AI-driven analytics, and enterprise governance capabilities, the platform can significantly improve organizational vulnerability management and cyber risk mitigation strategies.

VIII. CONCLUSION

The AI-Powered Regulatory Compliance Checker for Contracts provides an advanced and intelligent solution for automating the process of legal contract analysis and compliance verification. With the increasing complexity of regulatory standards and the growing volume of contractual documents in modern organizations, traditional manual review methods are no longer sufficient to ensure fast, accurate, and scalable compliance management. Manual approaches are time-consuming, costly, and vulnerable to human errors, which may lead to legal disputes, financial losses, operational failures, and reputational damage.

The proposed system successfully integrates Artificial Intelligence, Natural Language Processing, Machine Learning, and Large Language Models to overcome these limitations. By using NLP techniques such as tokenization, named entity recognition, clause extraction, semantic analysis, and contextual understanding, the system can automatically identify important legal clauses and evaluate them against regulatory requirements such as GDPR, HIPAA, PCI-DSS, SOX, and organizational policies. The inclusion of Large Language Models further enhances the system's capability to understand complex legal language, hidden meanings, and contextual relationships between contractual terms.

The research demonstrates that the proposed AI-based compliance checker significantly improves the speed, efficiency, consistency, and accuracy of contract analysis compared to traditional manual methods. The system not only detects non-compliant clauses and missing requirements but also performs intelligent risk analysis by categorizing legal, financial, operational, privacy, and reputational risks. The automated report generation feature helps legal teams and compliance officers make faster and more informed decisions.

In conclusion, the AI-Powered Regulatory Compliance Checker for Contracts represents a significant advancement in LegalTech and intelligent compliance management. The proposed system offers a practical, scalable, and efficient approach for automating contract review and regulatory verification. As AI technologies continue to evolve, automated compliance systems will become increasingly important for organizations seeking to reduce legal risks, improve governance, and enhance operational efficiency in the digital era.advancements.

REFERENCES

[1] OWASP Foundation, "OWASP Top 10: Web Application Security Risks," OWASP Foundation, 2021.

[2] R. Kissel, "Glossary of Key Information Security Terms," National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, Tech. Rep., 2013.

[3] P. Mell and K. Scarfone, "A Complete Guide to the Common Vulnerability Scoring System Version 3.1," FIRST Organization, 2019.

[4] S. Dua and X. Du, *Data Mining and Machine Learning in Cybersecurity*, Boca Raton, FL, USA: CRC Press, 2016.

[5] A. Mishra and P. Sharma, "Artificial Intelligence Techniques in Cybersecurity: A Survey," *IEEE Access*, vol. 10, pp. 45812–45834, 2022.

[6] D. P. Acharjya and K. Ahmed, "A Survey on Big Data Analytics: Challenges, Open Research Issues and Tools," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 2, pp. 511–518, 2016.

[7] OWASP Foundation, "Cross Site Request Forgery (CSRF)," OWASP Cheat Sheet Series, 2022.

[8] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, 3rd ed. Indianapolis, IN, USA: Wiley, 2020.

[9] Tenable Inc., "Nessus Vulnerability Scanner," Tenable Documentation, 2023.

[10] Greenbone Networks, "OpenVAS Scanner Documentation," Greenbone Security Manager, 2022.

[11] FIRST Organization, "CVSS v3.1 Specification Document," Forum of Incident Response and Security Teams, 2019.

[12] S. Frei, M. May, U. Fiedler, and B. Plattner, "Large-Scale Vulnerability Analysis," in *Proc. ACM SIGCOMM Workshop on Large-Scale Attack Defense*, 2006, pp. 131–138.

[13] M. Bishop, *Computer Security: Art and Science*, 2nd ed. Boston, MA, USA: Addison-Wesley, 2018.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, Cambridge, MA, USA: MIT Press, 2016.

[15] A. Souri and R. Hosseini, "A State-of-the-Art Survey of Malware Detection Approaches Using Data Mining Techniques," *Human-centric Computing and Information Sciences*, vol. 8, no. 3, pp. 1–22, 2018.

[16] S. Noel and S. Jajodia, "Managing Attack Graph Complexity Through Visual Hierarchical Aggregation," in *Proc. ACM CCS Workshop on Visualization and Data Mining for Computer Security*, 2004, pp. 109–118.

[17] M. Grinberg, *Flask Web Development: Developing Web Applications with Python*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2018.

[18] React Documentation Team, "React Developer Documentation," Meta Open Source, 2023.

[19] Todupunuri, A. (2024). Exploring the use of generative AI in creating deepfake content and the risks it poses to data integrity, digital identities, and security systems. Available at SSRN 5014688.

[20] Babburi, S. Lightweight Distributed Provenance Framework for Edge and IoT Data Systems.

[21] Gaddam, S. Integrating Analytics into the Development Process: Bridging the Gap between Data Insights and Design Execution.

[22] Immadi, S. K. (2025). Optimizing ERP for Human Capital Management. Applied Research for Growth, Innovation and Sustainable Impact, 377–384. https://doi.org/10.1201/9781003684657-63

[23] Poojari, R. INTELLIGENT SYSTEMS+B108 AND APPLICATIONS IN ENGINEERING.

[24] Poojari, R. Frameworks for Data Management and Lineage in Large-Scale Healthcare Data Systems.

[25] Poojari, R. Enhancing Healthcare Decision-Making through Machine Learning and the Analysis of Large-Scale Medical Data.

[26] Mahimalur, R. K., Vasgam, M., & Manoharan, D. Devops Lifecycle Management And Cloud Migration Assessments: A Security-Driven CICD Perspective.

[27] Purmani, S. S. R. (2025). Streamlining IT operations and service management with agile frameworks. European Journal of Advances in Engineering and Technology, 12(4), 76–81.

[28] Purmani, S. S. R. (2024). Aligning IT investment decisions with overall business strategy from an enterprise program management perspective, focusing on the integration of IT leadership in strategic decision-making processes. International Journal of Communication Networks and Information Security, 16(5), 1213–1219

[29] Cyril, H. P., & Kumara, S. (2026, February). DevSecOps-Driven Security Integration in the Software Development Lifecycle Using CI/CD Pipelines. In 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC) (pp. 1-6). IEEE.

[30] Kotte, G. (2025). Enhancing Cloud Infrastructure Security on AWS with HIPAA

Compliance Standards. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283660

[31] Kotte, G. (2025). Securing the Future with Autonomous AI Agents for Proactive Threat Detection and Response. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5283830

[32] Ranjbareslamloo, S., Dzukeya, G. A., Muhit, M. M. I., & Qattawi, A. (2025). Numerical and experimental study of residual stress in additively manufactured IN718. Manufacturing Letters, 44, 915–927. https://doi.org/10.1016/j.mfglet.2025.915927

[33] Viswanathan, V. (2024). Embedding Ethical Principles into Generative AI Workflows for Project Teams.

[34] Viswanathan, V. (2024). Pioneering Ethical AI Integration in Enterprise Workflows: A Framework for Scalable Team Governance. Available at SSRN 5375619.

[35] Mudusu, S. K. (2026, April 15). The secure intelligence framework: Architecting AI systems for a data-driven world. CIO (Foundry Expert Contributor Network).

[36] Mudusu, S. K. (2025, December 22). Cognitive data architecture: Designing self-optimizing frameworks for scalable AI systems. CIO (Foundry Expert Contributor Network).

[37] Gajula, S. (2026, March). Two Pillars of Banking Intelligence: A Comparative Analysis of AI Techniques for Fraud Prevention and Churn Mitigation. In 2026 14th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.

[38] Gajula, S. (2025, December). Ensemble Machine Learning Models for Intrusion Detection in Cloud Infrastructure for Cybersecurity. In 2025 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics (ICoABCD) (pp. 1-6). IEEE.

[39] Maturi, S. Y. (2025). Vulnerabilities in the 802.11 Wireless Client Selection Mechanis.

[40] Maturi, S. Y. Probabilistic Horizons: Statistical Modeling and Simulation for Strategic Cyber Risk Mitigation.

[41] Chowdhury, A. K., Muhit, M. M. I., & Islam, M. M. (2023). A practical review to the marine maintenance practice in Bangladesh and a proposed way forward to an efficient, long-term and cost-effective solution. In Proceedings of the 13th International Conference on Marine Technology (MARTEC 2022). https://doi.org/10.2139/ssrn.4445071

[42] Manoharan, D. (2025). Healthcare EDI Transaction Lifecycles Embedded with a Multi-Layer Verification Framework to Ensure Referential Integrity.

[43] Manoharan, D. (2026). AI-Driven Anomaly Detection Models for Preventing Claims Denials and Revenue Leakage in Healthcare. Available at SSRN 6385759.

[44] Ravishankara, M. (2026, February). CircuChain: Disentangling Competence and Compliance in LLM Circuit Analysis. In SoutheastCon 2026 (pp. 1-7). IEEE.

[45] Doragacharla, V. R. (2023). Comprehensive Benchmarking Analysis of Auto Scaling Approaches in Cloud Native Streaming Pipelines During Flash Sales and Holiday Traffic Peaks. Available at SSRN 6566479.

[46] P. Venkata Ramana. (2024). AI-driven predictive analytics in ERP systems for proactive supply chain optimization. International Journal of Innovative Engineering and Management Research (IJIEMR).

[47] Kumar Adabala, P. (2021). Optimizing ERP Modernization: A Smart Data Migration Framework Approach. International Journal of Enhanced Research in Science, Technology &Amp; Engineering, 10(07), 61–72. https://doi.org/10.55948/ijerste.2021.0708

[48] Pavan Kumar Adabala. (2026). Smart Retail Fuel Systems: IoT-Enabled Solutions for Loss Prevention and Environmental Safety. Computer Fraud and Security, 868–875. https://doi.org/10.52710/cfs.995

[49] Kavuri, S. (2025). Critical Review of Software Testing Problems in the Current Decade. International Journal on Science and Technology, 16(2). https://doi.org/10.71097/ijsat.v16.i2.9469

[50] Srikanth Kavuri. (2023). Machine Learning Approaches for Security Vulnerability Detection in Software Testing. Computer Fraud and Security. https://doi.org/10.52710/cfs.837

[51] Venkata Pavan Kumar Gummadi. (2023). MuleSoft Batch Processing: High-Volume Streaming Architecture. Computer Fraud and Security, 50–57. https://doi.org/10.52710/cfs.886

[52] Venkata Pavan Kumar Gummadi. (2026). Infrastructure Optimization Techniques for Enterprise Integration Platforms: A Comprehensive Analysis. Computer Fraud and Security, 37–44. https://doi.org/10.52710/cfs.875

[53] Venkata Pavan Kumar Gummadi. (2024). API Design and Implementation: RAML and OpenAPI Specification. Journal of Electrical Systems, 16(4), 76–85. https://doi.org/10.52783/jes.9329

[54] Venkata Pavan Kumar Gummadi. (2025). MuleSoft's Role in Advancing Sustainable Digital Infrastructure: An Enterprise Integration Perspective. Journal of Information Systems Engineering and Management, 10(62s), 1313–1321. https://doi.org/10.52783/jisem.v10i62s.13783

[55] Gummadi, V. P. K. (Ed.). (2025). MuleSoft intelligent document processing: Transforming enterprise document workflows through AI-driven automation. Journal of Computational Analysis and Applications, 34(12). https://doi.org/10.48047/jocaaa.2025.34.12.16

[56] Subramanian, V. K., Bhambri, S., & Gajula, S. (2026). Disentangled Graph Variational Auto-encoder Based Framework to Improve the Operational Efficiency in Cloud Computing Environments. Computer Vision and Robotics, 396–407. https://doi.org/10.1007/978-3-032-14044-9_32

[57] Gajula, S., & Margam, M. (2026). A Secure and Scalable Cloud-Based Banking Service Model Leveraging AI and Advanced Cyber Security. 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC), 1–5. https://doi.org/10.1109/icaic67076.2026.11395704

[58] Gajula, S. (2025). AI-Powered Forecasting Models, Optimizing Working Capital, Supply Chain Financing. 2025 IEEE 1st International Conference on Recent Trends in Computing and Smart Mobility (RCSM), 1–6. https://doi.org/10.1109/rcsm67767.2025.11507813