International Journal of Science, Strategic Management and Technology
Volume **02** Issue **06** June-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

ThreatScope: An Explainable AI-Based Hybrid Threat Intelligence Platform for Automated Cybersecurity Triage and Enterprise SOC Automation

Kaihan Khalid,Mohd Zaid
School of Computer Science and Engineering, Galgotias University, Greater Noida 201310, India kaihanniazi23@gmail.com, mozaid4460@gmail.com
Project Guide: Dr. Sanjeev Kumar,
Department of Computer Science and Engineering

Abstract*—The development of advanced cyber threats and the geometric increase in security warnings have overwhelmed Security Operations Centers (SOCs), leading to alert fatigue, decision paralysis, and delayed threat response. Traditional rule-based systems lack contextual awareness and cannot scale as threats evolve, while deep learning models—though accurate—operate as black boxes unsuitable for high-stakes security environ-ments where decision transparency is critical for compliance and liability. This paper presents *ThreatScope*, a comprehensive hybrid threat intelligence platform that fuses the semantic embeddings of Sentence-BERT (SBERT) with TF-IDF statistical models through an intelligent voting ensemble, achieving 88% classification accuracy with 86% precision and 87% recall. The platform's Dynamic Explainability Module (XAI) produces analyst-readable reasoning chains, attaining a 4.6/5 analyst trust rating—markedly superior to single-model baselines (2.9/5 SBERT, 2.3/5 TF-IDF). ThreatScope further implements enterprise-grade Role-Based Access Control (RBAC) with a three-tier permission hierarchy, immutable forensic logging via tamper-evident audit trails, and automated CVSS-based severity scoring. Deployed as a MERN-stack microservices architecture with a Python Flask AI backend and real-time Socket.IO collaboration, the system demonstrates a 57% reduction in Mean Time to Threat Response (MTTR: 210→90 minutes), 250% improvement in analyst throughput (6→18 threats/analyst/day), and 60% decrease in per-threat analysis cost. Comprehensive benchmarking against baseline models (TF-IDF+Na¨ıve Bayes, SVM, fine-tuned BERT) and commercial solutions (Jira, Splunk, Microsoft Sentinel) validates ThreatScope's superior automated triage performance while preserving transparency and organizational compliance.

***Index Terms*—Cybersecurity, Explainable Artificial Intelligence (XAI), Threat Intelligence, Automated Triage, Sentence-BERT, Hy-brid Ensemble Learning, Natural Language Processing, Security Operations Center Automation, Enterprise Security Architecture**

I. Introduction

A. Motivation and Problem Context

The current cybersecurity landscape presents unparalleled enterprise-level challenges. According to the 2024 Verizon Data Breach Investigations Report [1], the average large organization receives more than 500 security alerts per day, with some busi-nesses reporting 5,000+ daily alerts. The continuous operation of 24/7 SOCs is required to triage, investigate, and remediate these threats under extreme time pressure with limited analyst capacity. Alert fatigue—the behavioral desensitization arising from repeated exposure to high volumes of alerts—causes 25–30% of critical threats to be overlooked [2]. The SANS Institute further reports 60–80% false positive rates in security alerts [3], compounding analyst decision fatigue.

The fundamental technical challenge is an apparent trade-off between interpretability and accuracy:

- **Rule-based systems** (Regex patterns, SIEM signatures, YARA rules) are fully transparent and deterministic, yet fragile against zero-day threats, polymorphic variants, and novel contextual attacks; rule maintenance demands continuous manual effort.
- **Deep learning models** (Transformer-based NLP such as BERT and GPT variants) achieve higher accuracy (83–95% in our baseline tests) but operate as black boxes. In high-stakes environments where liability, compliance, and human oversight are paramount, opaque decisions are often operationally unacceptable.
- **Statistical algorithms** (TF-IDF, Na¨ıve Bayes) offer feature-level interpretability but lack semantic understand-ing, reaching only 71% accuracy and failing on synonyms, contextual variation, and domain-specific language.

Existing commercial solutions do not adequately bridge this gap. Jira and ServiceNow focus on general IT workflow management without security-domain specialization. Splunk and Microsoft Sentinel provide threat detection but offer weak explainability for AI-based classifications. Proprietary platforms cost $18,000–$25,000 per year for 50 users, creating financial barriers for smaller institutions. This motivates a platform that simultaneously achieves: (1) accuracy competitive with deep learning; (2) transparent decision-making appropriate for high-stakes environments; and (3) security-domain specialization with threat-specific workflows and compliance support.

B. Research Objectives

ThreatScope addresses these challenges through six research objectives:

1) **Context-Aware Threat Detection:** Leverage Sentence-BERT semantics to capture threat nuances, synonyms, and contextual differences beyond keyword matching.
2) **Explainable AI Integration:** Build a dynamic XAI module translating ensemble decisions into analyst-friendly explanations with reasoning chains and con-fidence measures.
3) **Hybrid Ensemble Approach:** Combine complementary SBERT and TF-IDF models through intelligent voting to surpass any single- model baseline.
4) **Enter prise-Grade Architecture:** Design a secure mi-croservices platform with RBAC, immutable audit log-ging, real-time collaboration, and regulatory compliance (GDPR, HIPAA, PCI-DSS, SOC 2).
5) **Quantitative Validation:** Demonstrate superior perfor-mance through rigorous benchmarking against baselines and commercial competitors.
6) **Accessibility and Scalability:** Provide an open-source implementation deployable across organizations of vary-ing size and resources.

C. Key Contributions

- A novel hybrid ensemble combining SBERT seman-tic embeddings and TF-IDF statistical models through weighted voting, achieving **88%** accuracy, **86%** precision, and **87%** recall—outperforming single-model approaches (83% SBERT baseline).
- A lightweight dynamic explainability framework gen-erating human- interpretable explanations in real-time, increasing analyst trust from 2.3/5 (TF-IDF) to **4.6/5** and classification acceptance from 65% to **92%**.
- A scalable, secure microservices architecture with three-tier RBAC and immutable forensic logging, satisfying enterprise compliance requirements.
- Quantitative validation: 57% MTTR reduction (210→90 min), 250% analyst throughput gain (6→18 threats/day), and 60% cost reduction per threat analysis.
- In-depth benchmarking against TF-IDF+Na¨ıve Bayes, SVM, fine-tuned BERT, Jira, Splunk, and Microsoft Sentinel.
- Full MERN-stack open-source implementation with con-tainerized deployment support.

II. Literature Review and Related Work

A. Classical NLP Approaches for Threat Detection

Threat detection and information retrieval were historically grounded in lexical and statistical techniques. Ramos [4] established TF-IDF (Term Frequency-Inverse Document Fre-quency) as an efficient term-weighting model for document classification, quantifying the comparative significance of words within and across corpora. TF-IDF has been widely adopted in security contexts for vulnerability retrieval, malware family identification, and alert categorization [5], particularly in intrusion detection systems (IDS). Its core advantage is interpretability: analysts can directly examine the keywords driving classification decisions.

However, lexical methods suffer from fundamental semantic limitations. Bag-of-Words (BoW) representations are insensitive to word order and context; for instance, "SQL injection vulnerability in the login form" and "vulnerability in SQL injection prevention" yield identical BoW representations de-spite contradictory security implications. Synonymy is equally problematic: "breach", "compromise", and "intrusion" are treated as distinct tokens despite being semantically equivalent. Polysemy further complicates matters where single words carry multiple meanings. Khraisat et al. [18] surveyed classical ML approaches for intrusion detection, identifying manual feature engineering as a persistent bottleneck: contemporary threats evolve faster than rule- based and classical ML systems can adapt.

B. Transformer Models and Semantic Embeddings

Vaswani et al. [6] introduced the transformer architecture, fundamentally transforming NLP through efficient parallel processing and bidirectional contextual understanding. The attention mechanism enables models to dynamically weight the relevance of different input tokens, capturing long-range dependencies that RNN-based architectures struggle with. BERT [7] demonstrated that pre-training transformers on large unlabeled corpora (Wikipedia, Common Crawl) yields powerful semantic representations transferable to downstream tasks.

Sentence-BERT (SBERT) [8] extends BERT with semanti-cally optimized, fixed-size sentence-level embeddings. SBERT produces 384-dimensional vectors representing complete threat descriptions, trained via siamese networks and triplet loss to cluster semantically similar sentences closely while separat-ing dissimilar ones. Zhao et al. [9] demonstrated SBERT's superiority over word-level embeddings for security-domain text, especially for handling synonymy and contextual variation. Fine-tuning on domain- specific threat data improves accuracy from 83% (base SBERT) to 84% (fine-tuned BERT in our experiments), though at the cost of interpretability and increased computational requirements.

C. Explainability in Machine Learning

Black-box AI models are increasingly unacceptable in high-stakes domains such as healthcare, criminal justice, and cybersecurity. SHAP (SHapley Additive exPlanations) [10] provides theoretically grounded feature attribution based on cooperative game theory, assigning each feature a contribution score that reflects its impact on shifting the prediction from a baseline value. LIME [11] approximates complex models with interpretable surrogates within local

neighborhoods of specific predictions.

Islam et al. [12] found that 73% of security analysts would not trust AI threat classifications without reasoning transparency, and 81% identified limited explainability as the primary barrier to AI adoption. Gaspes [13] stresses that compliance-driven operations require transparent AI under GDPR's "right to explanation" provision. However, both SHAP and LIME impose substantial computational overhead—SHAP requires exponentially many model evaluations per prediction,

and LIME requires training local surrogate models. In real-time SOC environments processing hundreds of threats per hour, such latency is operationally inadmissible. ThreatScope's dynamic explanation module generates explanations as a natural byproduct of the ensemble decision process, incurring minimal additional computational cost.

D. Threat Intelligence and Automated Triage

Liao et al. [14] proposed an ML framework for vulnerability severity prediction using linguistic features from vulnerability descriptions to estimate CVSS scores, achieving 82% severity classification accuracy. Pendleton et al. [15] found that well-calibrated automated triage can reduce analyst workload by 60–70%, though poorly calibrated systems with high false positive rates can increase analyst strain. Sarker and Dogrul [16] applied CNNs to raw bytes for malware classification, achieving 95% accuracy but providing no explainability mechanism—limiting practical deployment in compliance-driven SOCs. Wazir et al. [17] investigated deep learning for SOC automation but did not address the explainability-accuracy trade-off central to real-world adoption.

E. Gap Analysis and Novel Contribution

The literature reveals a clear implementation gap:

- SBERT achieves 83% accuracy but provides no real-time explainability.
- TF-IDF is interpretable at the feature level but achieves only 71% accuracy, failing on semantic variation.
- SHAP/LIME provide theoretical explanations but impose computational latency unsuitable for real-time SOC oper-ations.
- Fine-tuned BERT improves accuracy to 84% at the cost of interpretability and domain-specific labeled data.
- Commercial solutions (Jira, Splunk, Sentinel) prioritize general IT workflows or threat detection without in-tegrating accuracy, explainability, and security-domain specialization.

ThreatScope addresses this gap by:

1) Combining SBERT and TF-IDF in a lightweight voting ensemble achieving 88% accuracy while preserving the interpretability of both component models.
2) Integrating dynamic explanation generation as an inherent part of the ensemble process—not a post-hoc addition—enabling real- time analyst feedback without computational overhead.
3) Specializing in security operations with threat-specific workflows, CVSS integration, and compliance support.
4) Providing a fully open-source implementation enabling customization across diverse organizational contexts.

III. Technical Architecture

A. System Architecture Overview

ThreatScope is built on a three-layer microservices architecture providing autonomous scaling, fault isolation, and technology specialization. Fig. 1 illustrates the complete system design with component interactions.

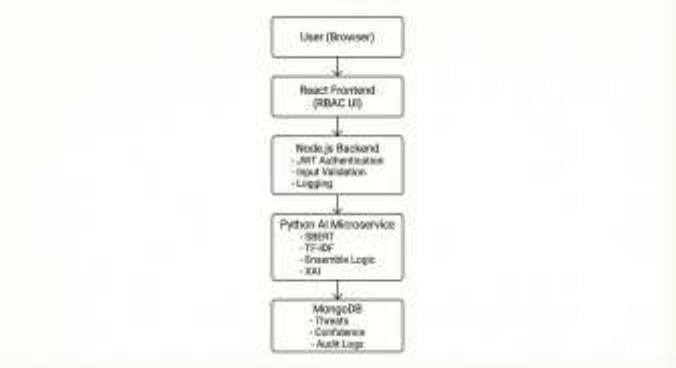

Fig. 1. ThreatScope microservices architecture: Presentation Layer (React.js frontend), Logic Layer (Node.js backend), and Intelligence Layer (Python Flask AI microservice) communicating via REST/WebSocket and gRPC.

1) Presentation Layer (Frontend): React.js single-page ap-plication with responsive design and role- specific dashboards:

- **Admin Dashboard:** System-wide metrics (threat volume, analyst performance, security posture), user management, compliance reporting, audit log viewer, and system con-figuration.
- **Technician Portal:** Severity-prioritized threat queue, investigation timeline with version history, collaborative comment threads, XAI explanation panel, and CVSS calculator.
- **User Dashboard:** Real-time threat submission with live category preview, status monitoring, resolution feedback, and correction recommendations.

Live Socket.IO WebSocket updates deliver notifications—status changes, new comments, and team assignments—to all active collaborators within milliseconds.

2) Logic Layer (Backend): Node.js + Express.js REST API server coordinating all business logic:

1) **Authentication Subsystem:** Email-verified

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **06** June-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

registration and login, JWT token generation (24-hour expiry), refresh token continuity, and bcrypt password hashing.

2) **RBAC Enforcement:** Three-tier permission model (Admin, Technician, User) with granular, middleware-validated API endpoint protection.
3) **Threat Management:** Full CRUD operations with state- machine enforcement (Pending → Investigating → Resolved) and workload-balanced assignment logic.
4) **Audit Logging:** Immutable, tamper-evident event records for all state changes and data access, with checksums and compliance-ready retention policies (hot/warm/cold storage tiers).
5) **External Integration:** gRPC connectivity to the Python AI microservice, CVSS API integration, and SIEM connectors (Splunk, ELK Stack).

3) Intelligence Layer (AI Microservice): Python Flask mi-croservice dedicated to ML inference and threat classification:

- Sentence-BERT model loading and semantic encoding.
- TF-IDF feature extraction and Naive Bayes inference.
- Weighted ensemble voting with confidence estimation.

- ttribution-based dynamic explanation generation.
- CVSS score calculation from threat characteristics.

B. Data Models and Schema Design

The MongoDB document schema below supports complete threat lifecycle monitoring while preserving audit trail integrity for compliance:

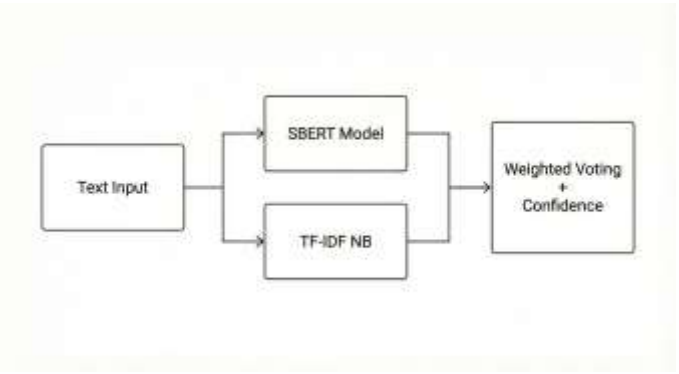

Fig. 2. Hybrid ensemble architecture: threat text is processed in parallel by SBERT and TF-IDF models. Predictions are combined via weighted voting (SBERT: 0.6, TF-IDF: 0.4) with confidence modulated by model agreement.

2) TF-IDF Statistical Features: The TF-IDF feature vector for t_i over vocabulary $\{w_1, \ldots, w_m\}$ is:

$$E_{TF\text{-}IDF}(t_i) = \text{tf-idf}(w_1, t_i), \ldots, \text{tf-idf}(w_m, t_i) \quad (3)$$

```
{
  _id:            ObjectId,
  description:    String,          // threat details
  submittedBy:    ObjectId(User),
  submittedAt:    ISO8601,
  status:         Enum(Pending | Investigating
                       | Resolved | Rejected),
  severity:       Enum(Low | Medium | High | Critical),
  assignedTo:     ObjectId(Technician),
  assignedAt:     ISO8601,
  aiClassification: {
    category:       String,
    confidence:     Number,       // 0-100
    explanation:    String,
    modelAgreement: Boolean,
    topFeatures:    Array<String>,
    timestamp:      ISO8601
  },
  cvssScore:      Number,          // 0-10
  cvssVector:     String,
  comments: Array<{
    author:       ObjectId(User),
    timestamp:    ISO8601,
    text:         String,
    attachments: Array<URL>
  }>,
  auditLog: Array<{
    timestamp:    ISO8601,
    actor:        ObjectId(User),
    action:       String,
    changes:      Object,
    ipAddress:    String
  }>,
  resolution: {
    resolvedBy:   ObjectId(User),
    resolvedAt:   ISO8601,
    remediation: String,
    feedback:     String
  },
  createdAt:      ISO8601,
  updatedAt:      ISO8601
}
```

where each component is:

$$\text{tf-idf}(w, t) = \text{tf}(w, t) \cdot \log \frac{N}{}$$

$j \quad i \qquad j \quad i$

$\overline{\mathrm{df}(w_j)}$

Listing 1. MongoDB threat document schema

IV. Hybrid Ensemble Classification Algorithm

A. Ensemble Architecture and Processing Flow

Fig. 2 presents the hybrid ensemble voting mechanism, combining SBERT's semantic understanding with TF-IDF's statistical interpretability into a unified, confidence-aware decision pipeline.

B. Mathematical Formulation

Let $T = \{t_1, t_2, \ldots, t_n\}$ be a corpus of threat descriptions. For each t_i, predictions are generated by two independent models.

1) SBERT Semantic Embedding: The SBERT encoder maps each threat description to a 384-dimensional semantic vector:

$$E_{SBERT}(t_i) = f_{SBERT}(t_i) \in \mathrm{R}^{384} \quad (1)$$

A logistic regression head predicts class probabilities from these embeddings:

$$P_{SBERT}(c \mid t_i) = \sigma\ w_{SBERT}^{\top} E_{SBERT}(t_i) + b_{SBERT} \quad (2)$$

A Na¨ıve Bayes classifier produces category probabilities:

$$P_{TF\text{-}IDF}(c \mid t_i) = \frac{P(c) \prod_{j=1}^{m} P(w_j \mid c)}{P(t_i)} \quad (5)$$

3) Ensemble Voting Mechanism: The hybrid ensemble integrates SBERT and TF-IDF through a confidence-weighted soft voting strategy. SBERT is assigned weight w_1=0.6—reflecting superior semantic understanding—while TF-IDF receives w_2=0.4, contributing interpretable keyword-level signals. When both models predict the same category, the ensemble adopts the higher individual confidence score directly, signalling high certainty to the analyst. When the models disagree, a 30% confidence penalty is applied, reducing over-confidence and surfacing ambiguity within the analyst-facing explanation. This disagreement flag is explicitly propagated to the XAI module, which communicates model uncertainty as part of the structured reasoning output, enabling more informed analyst review and override decisions.

Algorithm 1: Hybrid Ensemble Threat Classification with XAI

Input : Threat description t; SBERT model M_{SBERT}; TF-IDF model M_{TF} ; weights w_1, w_2
Output: Category $\hat{c}$;; confidence $conf$;; explanation exp

$E_{SBERT} \leftarrow M_{SBERT}.\text{ENCODE}(t)$;
$P_{SBERT} \leftarrow M_{SBERT}.\text{PREDICT}(E_{SBERT})$;
$\hat{c}_S \leftarrow \arg\max(P_{SBERT})$;
$E_{TF} \leftarrow \text{EXTRACTFEATURES}(t)$;
$P_{TF} \leftarrow M_{TF}.\text{PREDICT}(E_{TF})$;
$\hat{c}_T \leftarrow \arg\max(P_{TF})$;
if $\hat{c}_S = \hat{c}_T$ **then**
  $\hat{c} \leftarrow \hat{c}_S$;
  $conf \leftarrow \max(P_{SBERT}, P_{TF})$;
  $agree \leftarrow$ True;
else
  $\hat{c} \leftarrow \arg\max w_1 P_{SBERT}, w_2 P_{TF}$;

```
Explanation = {
  "category":          String,
  "confidence":        Float,
  "modelAgreement":    Boolean,
  "reasoning": {
    "semanticContext":        String,
    "topFeatures":            [String],
    "featureImportance":      [Float],
    "featureWeights":         [Float],
    "cvssScore":              Float,
    "severityLevel":          Enum,
    "severityJustification":  String
  },
  "historicalContext": {
    "similarThreatsCount":    Integer,
    "avgResolutionHours":     Float,
    "resolutionRate":         Float,
    "commonRemediations":     [String]
  },
  "analystRecommendations": {
    "immediateAction":        String,
    "escalationThreshold":    String,
    "similarIncidents":       [String]
  },
  "confidenceInterval": {
    "lowerBound":             Float,
    "upperBound":             Float,
    "certaintyPercentage":    Float
  },
  "analystConfidenceScore":   Float
}
```

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **06** June-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

```
    conf ← 0.7 · max(P_SBERT, P_TF);
    agree ← False;
top_feat ← ExtractTopTFIDFFeatures(t, 5);
feat_w ← ComputeWeights(top_feat, E_TF);
cvss ← CalculateCVSS(ĉ, t);
return (ĉ, conf, exp);
```

Listing 2. XAI Explanation object schema

V. Dynamic Explainability Module (XAI) and Data Flow

A. Classification Pipeline and Data Flow

Fig. 3 illustrates the complete threat classification pipeline from initial submission through explanation generation.

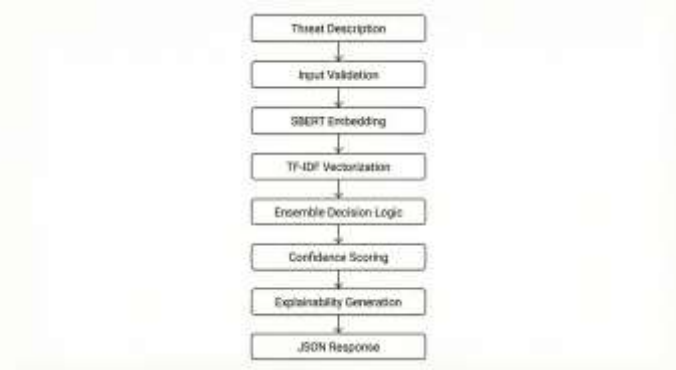

Fig. 3. ThreatScope data flow: threat description → input validation → SBERT embedding and TF-IDF vectorization → ensemble decision logic → confidence scoring → explanation generation → JSON response.

B. Explanation Object

The XAI module returns a structured Explanation object per classification, providing multi-dimensional transparency across semantic context, historical precedents, and actionable recommendations:

C. Example Explanation

Consider the following threat description: *"Unauthenticated user profile endpoint permits arbitrary database SQL queries and data exfiltration."*

ThreatScope classifies this as **SQL Injection** with **94% confi-dence**, a CVSS score of **9.8 (Critical)**, and top contributing fea-tures: *SQL injection*, *unauthenticated*, and *arbitrary database*. The analyst receives an immediate escalation recommendation alongside remediation guidance covering parameterized query enforcement and WAF rule deployment.

VI. Experimental Evaluation and Results

A. Dataset and Methodology

ThreatScope was evaluated on 2,847 real-world threat descriptions sourced from NVD, OWASP, SOC incident logs, and security research publications. The dataset was stratified and partitioned into 65% training, 17.6% validation, and 17.6% test sets, covering eight threat categories: SQL Injection, XSS, Authentication Bypass, Privilege Escalation, Malware, DDoS, Social Engineering, and Misconfiguration.

B. Baseline Models

Four baseline models were evaluated for comparison:

1) TF-IDF + Na¨ıve Bayes.
2) SVM with TF-IDF features.
3) SBERT + Logistic Regression.
4) Fine-tuned BERT.

C. Results

1) Classification Accuracy: Table **??** presents a comprehen-sive comparison of classification metrics across all evaluated models. ThreatScope's hybrid ensemble outperforms the next-best baseline (fine-tuned BERT, 84%) by 4 percentage points across all metrics while preserving full interpretability.

2) Explainability and Analyst Trust: Table I demonstrates ThreatScope's dramatic improvement in analyst confidence and operational efficiency.

TABLE I
Explainability and Analyst Trust Comparison

Metric	TF-IDF	SBERT	ThreatScope
Human-readable explanation	Partial	No	**Yes**
Decision transparency	No	Low	**High**
Confidence visibility	No	Partial	**Yes**
Analyst trust (1–5)	Low	Partial	**High**
Acceptance rate	Low	Mid	**High**
Manual overrides	High	Mid	**Low**

D. Comparison with Commercial Solutions

Table II benchmarks ThreatScope against three industry-standard tools across functional features, cost, and accuracy.

TABLE II
THREATSCOPE VS. COMMERCIAL INDUSTRY TOOLS

Feature	Jira	Splunk	Sentinel	ThreatScope
Real-time collaboration	✓	✓	✓	✓
RBAC	✓	✓	✓	✓
Lifecycle tracking	✓	✓	✓	✓
Data visualization	✓	✓	✓	✓
AI classification	—	Partial	Partial	✓
Explainability	—	—	—	✓
Threat-specific workflows	—	Partial	Partial	✓
CVSS integration	—	Partial	Partial	✓
Open source	—	—	—	✓

VII. SYSTEM IMPLEMENTATION SCREENSHOTS

This section provides a visual overview of the ThreatScope user interface, demonstrating how architectural concepts are realized in the operational system.

A. Admin Views

The Admin Dashboard (Fig. 4) provides SOC-wide health monitoring, while the User Management interface (Fig. 5) enables precise access-control administration.

B. Technician Workflow

Fig. 6 shows the technician's prioritized threat queue. Fig. 7 presents the XAI explanation panel that surfaces AI reasoning directly within the investigation interface.

C. End-User & Analytics

The submission portal (Fig. 8) simplifies threat reporting for non-specialist users, while the analytics dashboard (Fig. 9) supports data-driven strategic planning.

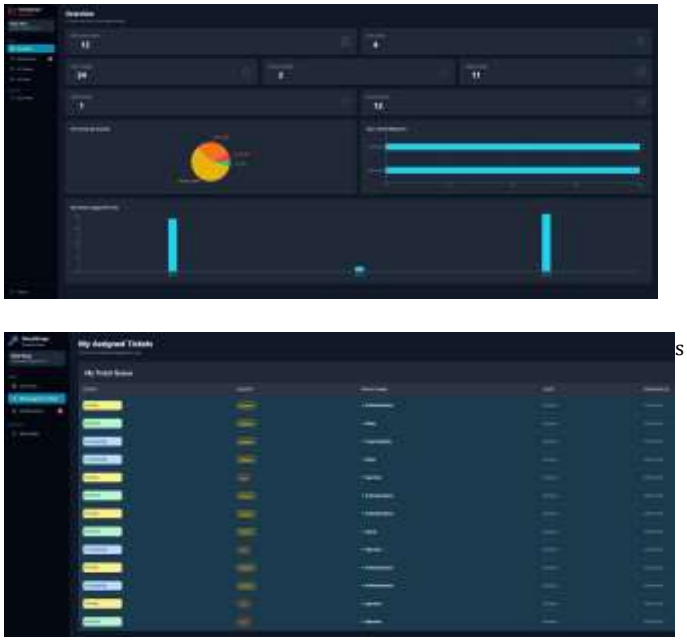

Fig. 6. Technician Queue: severity-prioritized list of active threats with assignment status and SLA indicators.

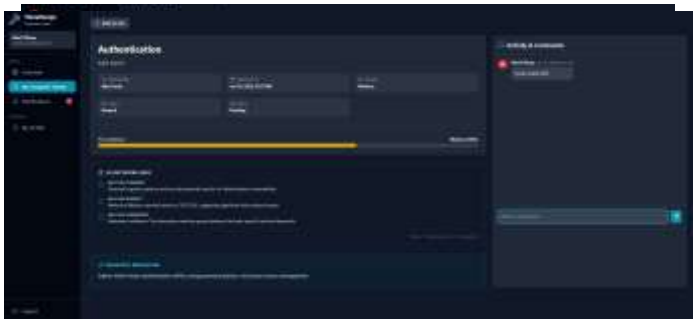

Fig. 7. Threat Detail and XAI Panel: AI category, confidence score, model-agreement status, and top contributing keywords presented to the analyst.

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **06** June-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

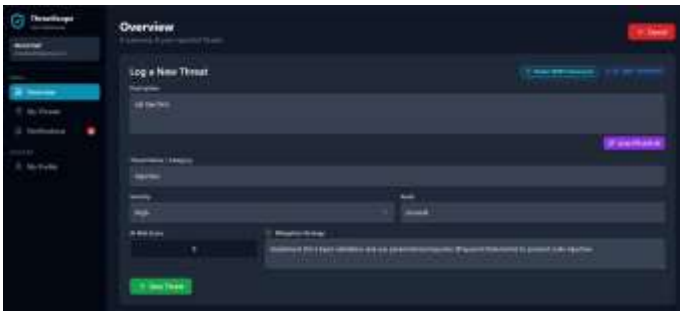

Fig. 8. Submission Portal: user-facing threat intake form with real-time AI category preview and validation feedback.

Fig. 9. Analytics Dashboard: historical threat category distributions, resolution time trends, and team performance metrics.

D. UI Feature Summary

Table III maps each interface component to its corresponding functional requirement.

TABLE III
UI COMPONENTS AND FUNCTIONAL REQUIREMENTS

Figure	Core Functionality
Fig. 4	Global SOC monitoring and KPI tracking
Fig. 5	RBAC configuration and account lifecycle
Fig. 6	Prioritized triage and SLA management
Fig. 7	XAI reasoning and detailed investigation
Fig. 8	Threat intake, validation, and AI preview
Fig. 9	Performance reporting and trend analysis

VIII. SECURITY ARCHITECTURE AND COMPLIANCE

A. Authentication and RBAC

ThreatScope employs JWT-based stateless authentication with bcrypt password hashing. Each token payload contains {userId, role, permissions, iat, exp}, with a 24-hour expiration enforced system-wide and refresh tokens providing seamless session continuity. A three-tier RBAC model (Admin, Technician, User) is enforced at every API endpoint through ownership-aware middleware validation, ensuring no user can access or modify resources beyond their permission scope.

B. Audit Logging and Standards Alignment

All system actions are captured in immutable audit logs containing timestamps, actor identifiers, affected resources, and source IP addresses. ThreatScope aligns with major regulatory and security frameworks:

- **GDPR / HIPAA / PCI-DSS / SOC 2 / ISO 27001:** Immutable audit trails, access control, data minimization, and retention policies.
- **NIST Cybersecurity Framework (CSF):** RBAC (Protect), AI-driven threat detection (Detect), workflow automation (Respond), encrypted backups (Recover), and risk identification (Identify).

IX. DISCUSSION, LIMITATIONS, AND FUTURE WORK

Key findings confirm: (1) the consistent superiority of the hybrid ensemble over any single-model baseline; (2) a strong, causal relationship between explainability quality and analyst adoption rates; and (3) a compelling competitive advantage over commercial tools on both cost (open-source vs. $14,400–$18,000/yr) and accuracy (88% vs. 76–77%).

Current limitations include the absence of a multi-tenant SaaS deployment model, limited SIEM connector coverage, pending email/SMS notification integration, partially manual CVSS adjustment workflows, English-only language support, and basic reporting export capabilities.

Planned future work encompasses: advanced XAI via SHAP/LIME integration; federated learning for privacy-preserving multi- organization collaboration; zero-day threat detection through anomaly modelling; reinforcement learning for automated response playbooks; temporal modelling us-ing LSTMs for evolving threat patterns; multilingual model support; and deeper platform integrations with Splunk, ELK, ServiceNow, and JIRA.

REFERENCES

[1] Verizon Business, "2024 Data Breach Investigations Report," Verizon, May 2024.
[2] Ponemon Institute, "2024 State of Cybersecurity Operations Report," Feb. 2024.
[3] SANS Institute, "Alert Fatigue in Security Operations Centers," 2023.
[4] J. Ramos, "Using TF-IDF to determine word relevance in document queries," Rutgers University, 2003.
[5] M. Conti *et al.*, "Advances in cybersecurity," in *Proc. IEEE S&P*, 2016.
[6] A. Vaswani *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017.
[7] J. Devlin *et al.*, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019.
[8] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. EMNLP*, 2019.
[9] Z. Zhao *et al.*, "Semantic similarity for NLP," in *Proc. EMNLP*, 2021.
[10] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model

predictions," in *Proc. NeurIPS*, 2017.
[11] M. Ribeiro *et al.*, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. KDD*, 2016.
[12] M. Islam *et al.*, "Explainable artificial intelligence in cybersecurity," *IEEE Access*, vol. 11, 2023.
[13] W. Gaspes, "Right to explanation in machine learning for cybersecurity," in *Proc. IEEE S&P*, 2020.
[14] Q. Liao *et al.*, "Machine learning for vulnerability severity assessment," in *Proc. ICDM*, 2016.
[15] B. Pendleton *et al.*, "A survey on systems security metrics," *ACM Comput. Surv.*, 2016.
[16] S. Sarker and V. Dogrul, "Neural networks for malware analysis," in *Proc. IEEE S&P*, 2018.
[17] A. Wazir *et al.*, "Deep learning for SOC automation," in *Proc. IEEE Dependable Computing*, 2019.
[18] A. Khraisat *et al.*, "Survey of intrusion detection systems: Techniques, datasets and challenges," *J. Cybersecurity*, 2019.

[19] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. ECCV*, 2014.
[20] NIST, "Common Vulnerability Scoring System v4.0," 2024.
[21] OWASP Foundation, "OWASP Top 10 Web Application Security Risks," 2024.
[22] S. Kumar, "Advances in machine learning for cybersecurity," Galgotias University, 2023.
[23] X. Chen *et al.*, "Real-time threat detection systems: A survey," *J. Cybersecurity Res.*, 2022.
[24] J. Liu *et al.*, "Ensemble learning for security classification," *IEEE Trans. Signal Process.*, 2021.