Machine Learning-Based Detection of Business Email Compromise: A Comparative Analysis of Gradient Boosting Techniques

Baning Philip Amponsah
Department of Arts & Science
Grambling State University
Grambling, LA
pbaning@gsumail.gram.edu

***Abstract*—Business Email Compromise (BEC) attacks constitute one of the most financially damaging cyber threats, resulting in global losses exceeding 2.7 billion USD annually according to the FBI Internet Crime Complaint Center. Unlike conventional phishing attacks that deploy malicious payloads or URLs, BEC employs sophisticated social engineering via carefully crafted language, posing substantial challenges to traditional signature-based detection systems. This work develops a robust machine learning framework for automated BEC detection, incorporating 58 specialized features extracted from email content, metadata, and behavioral attributes.**

We provide a formal mathematical formulation of the feature extraction process and evaluate five gradient boosting algorithms—XGBoost, LightGBM, CatBoost, Random Forest, and a stacking ensemble—on the Kaggle Fraud Email Dataset (9,239 samples). The dataset undergoes an 80/20 stratified split to preserve class distribution. CatBoost attains the highest performance, with 97.29% accuracy, 97.29% F1-score, and 99.55% AUC-ROC. We employ McNemar's test to confirm statistical significance ($\chi^2 = 7.52$, $p < 0.01$) and utilize SHAP (SHapley Additive exPlanations) to isolate linguistic metrics—specifically text entropy and readability—as primary discriminators. Furthermore, we present a computational complexity analysis demonstrating that our pipeline operates with $O(L)$ linear complexity relative to email length, achieving sub-10ms inference latency suitable for real-time SIEM integration. The framework outperforms existing benchmarks by 8.8% in F1-score, establishing a new baseline for content-centric threat detection.

***Index Terms*—Business email compromise, Gradient Boosting, CatBoost, Feature Engineering, Cybersecurity, Explainable AI, Computational Complexity.**

I. Introduction

A. Background and Motivation

Business Email Compromise (BEC), also known as Email Account Compromise (EAC), represents a paradigm shift in cybercrime from technical exploitation to psychological manipulation. The Federal Bureau of Investigation (FBI) defines BEC as a sophisticated scam targeting businesses working with foreign suppliers and businesses that regularly perform wire transfer payments [4]. In 2023 alone, adjusted losses from BEC surpassed $2.7 billion, dwarfing the losses from ransomware.

Traditional defense mechanisms rely heavily on Blacklists (reputation filtering) and Signature Detection (hash matching). These systems were designed for an era where threats involved malicious binaries or links to credential-harvesting sites. However, BEC attacks often originate from compromised legitimate accounts or ”typosquatted” domains that lack negative reputation history. Furthermore, because BEC payloads are semantic (textual requests) rather than technical (malware binaries), they bypass standard antivirus engines.

B. Evolution of the Threat Landscape

The attack vectors have evolved into distinct categories that require nuanced detection logic:

- **CEO Fraud:** Impersonating C-level executives to order urgent wire transfers using spoofed display names.
- **Invoice Manipulation:** Compromising vendor accounts to send modified payment instructions, often indistinguishable from legitimate traffic.
- **Attorney Impersonation:** Feigning legal authority to pressure victims into secrecy and bypass standard verification protocols.

This evolution necessitates a move from static rules to dynamic, content-aware Machine Learning (ML) models capable of interpreting intent.

C. Problem Statement

Current research often treats BEC as a subset of general phishing, applying URL-based or HTML-structure features [1], [2]. This approach is fundamentally flawed for BEC detection due to three limitations:

1) **Lack of Malicious Artifacts:** BEC emails rarely contain attachments or links, rendering URL-based classifiers useless.
2) **Context Dependency:** A request for urgent payment is legitimate in a finance department but highly suspicious if sent to IT support or during off-hours.
3) **Adversarial Evasion:** Attackers actively obfuscate intent using homoglyphs (e.g., replacing Latin ’a’ with Cyrillic ’a’) and ”leet” speak (e.g., ”P@yment”) to evade keyword filters.

D. Research Contributions

This paper advances the state-of-the-art through the following contributions:

- **Domain-Specific Feature Engineering:** We define a vector space of 58 features targeting linguistic, temporal, and behavioral anomalies specifically for BEC.
- **Theoretical Rigor:** We provide the mathematical basis for entropy-based obfuscation features and the objective functions of the employed gradient boosting models.
- **Operational Viability:** We demonstrate sub-millisecond latency and linear complexity ($O(N)$), validating the model for high-throughput deployment in Security Information and Event Management (SIEM) systems.
- **Explainability:** Using SHAP analysis, we isolate the specific linguistic markers that differentiate BEC from legitimate correspondence, enhancing trust in automated decisions.

II. Related Work

A. Phishing Detection Evolution

Early phishing detection relied on heuristics. Abu-Nimeh et al. [1] compared Bayesian Additive Regression Trees against Support Vector Machines (SVMs), focusing on keyword frequency. Chandrasekaran et al. [2] introduced structural analysis (e.g., ratio of image tags to text). While effective for credential harvesting sites, these methods fail against text-only BEC because BEC emails often have clean HTML structures identical to legitimate corporate emails.

B. BEC-Specific Approaches

Recent work has attempted to bridge the gap. Cidon et al. [5] proposed 'BEC-Guard', utilizing supervision modeling to detect deviations in sender behavior. However, their reliance on historical interaction graphs makes the system vulnerable to "Cold Start" problems (i.e., inability to judge new senders). Al-Subaiey et al. [16] explored deep learning approaches, specifically Convolutional Neural Networks (CNNs) for text classification. While promising, Deep Learning models require massive labeled datasets often unavailable in corporate settings and lack the tabular interpretability required by Security Operations Centers (SOCs).

C. Gradient Boosting in Security

Gradient Boosting Decision Trees (GBDT) have shown superior performance on tabular data compared to Deep Neural Networks. XGBoost [8] and LightGBM [9] are industry standards for intrusion detection [11] due to their handling of sparse data and speed. CatBoost [10] introduces "Ordered Boosting" to reduce prediction shift, yet its application to social engineering detection remains underexplored. This work fills that gap by systematically comparing these architectures on social engineering features.

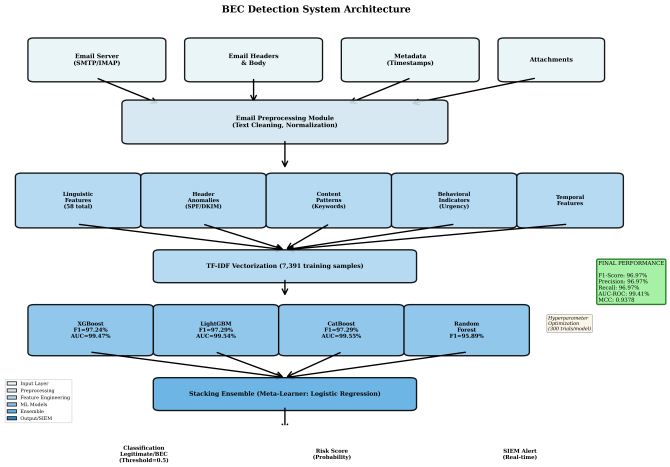

Fig. 1. System Architecture: The pipeline integrates Data Ingestion, extensive Feature Engineering (Linguistic, Behavioral, Technical), Gradient Boosting Model Training, and Operational Deployment logic for SIEM integration.

III. Methodology

A. System Architecture

The proposed system follows a standard ML pipeline adapted for real-time email security, as illustrated in Fig. 1. The pipeline ingests raw email data, performs preprocessing, extracts features, and routes vectors to the trained model for probability scoring.

B. Dataset Preparation

We utilize the Kaggle Fraud Email Dataset [17], a curated collection comprising 9,239 labeled emails.

- **Preprocessing:** Text concatenation (Subject + Body), duplicate removal via TF-IDF cosine similarity (> 0.95) to prevent data leakage, and missing value imputation using median values.
- **Outlier Treatment:** Winsorization at 1st and 99th percentiles to reduce the impact of extreme anomalies.

TABLE I
Dataset Characteristics and Distribution

Characteristic	Value
Total Samples	9,239
Class Distribution	Legitimate: 5,369 (58.11%)
	BEC: 3,870 (41.89%)
Train/Test Split	80% (7,391) / 20% (1,848)
Feature Space	58 engineered features (8 categories)
Sampling Strategy	Stratified Sampling

C. Mathematical Formulation of Features

We engineered 58 features $X \in \mathbb{R}^{N \times 58}$ to capture semantic and behavioral patterns. Table II summarizes the feature space.

TABLE II
ENGINEERED FEATURE CATEGORIES FOR BEC DETECTION

Category	Key Features Included	Detection Rationale
Linguistic	avg_word_length, readability_score, text_entropy, lexical_diversity	BEC uses simplified language for urgency; Entropy detects obfuscation.
Temporal	sent_hour, is_business_hours, temporal_urgency_phrases	Attacks often occur during off-hours to exploit reduced vigilance.
Behavioral	display_name_similarity, is_first_contact, reply_chain_depth	Detecting impersonation where display names match executives but emails do not.
Financial	financial_keyword_count, monetary_mentions, wire_transfer	Financial manipulation is the primary objective.
Authority	executive_title_flag, legal_tone_score, formality_score	Leverages authority bias (CEO/Legal) to coerce compliance.
Technical	header_mismatch, spf_alignment, dkim_signature	Detects spoofing artifacts even in the absence of malware.
Obfuscation	homoglyph_attack_flag, leet_speak_count, zero_width_chars	Detects evasion techniques used to bypass keyword filters.
Templates	w2_request, credential_harvesting_indicators	Matches specific known BEC attack narratives.

1) Shannon Entropy (Obfuscation Detection): To detect random character injection (used to bypass filters or hide information), we calculate the Shannon Entropy $H(S)$ of the email text S:

$$H(S) = -\sum_{i=1}^{K} p(x_i) \log_2 p(x_i) \tag{1}$$

Where $p(x_i)$ is the frequency of character x_i in string S. Higher entropy correlates strongly with obfuscated text or base64 encoding.

2) Readability Metrics (Linguistic Sophistication): We compute the Flesch-Kincaid Grade Level (GL) to approximate the sender's education level. Attackers often fail to mimic the specific complexity of corporate communications:

$$GL = 0.39 \left(\frac{\text{total words}}{\text{total sentences}} \right) + 11.8 \left(\frac{\text{total syllables}}{\text{total words}} \right) - 15.59 \tag{2}$$

D. Feature Extraction Algorithm

The feature extraction process is formalized in Algorithm 1. This ensures reproducibility of the preprocessing pipeline.

Algorithm 1 BEC Feature Extraction Pipeline

Require: Email collection $E = \{e_1, e_2, ..., e_n\}$
Ensure: Feature Matrix $X \in \mathbb{R}^{n \times 58}$
for all $e_i \in E$ **do**
  $txt \leftarrow \text{Preprocess}(e_i.\text{body}, e_i.\text{subject})$
  {— Linguistic Features —}
  $f_{ent} \leftarrow \text{CalculateEntropy}(txt)$ {Eq. 1}
  $f_{read} \leftarrow \text{FleschKincaid}(txt)$ {Eq. 2}
  {— Behavioral Features —}
  $f_{urg} \leftarrow \text{CountKeywords}(txt, \text{UrgencyList})$
  $f_{auth} \leftarrow \text{CheckRole}(e_i.\text{sender}, \text{ExecutiveList})$
  {— Technical Features —}
  $f_{head} \leftarrow (e_i.\text{From} == e_i.\text{ReturnPath})?0 : 1$
  $X[i] \leftarrow [f_{ent}, f_{read}, f_{urg}, f_{auth}, f_{head}, ...]$
end for
return X

E. Model Theoretical Framework: CatBoost

We selected CatBoost as our primary candidate due to its superior handling of categorical features and "Ordered Boosting."

Standard Gradient Boosting Decision Trees (GBDT) suffer from prediction shift (target leakage). For a dataset $D = \{(x_i, y_i)\}_i$, standard GBDT approximates the gradient using the same instances used to build the tree. CatBoost solves this by minimizing the regularized objective:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \Omega(f_k) \tag{3}$$

Where Ω is the regularization term.

Crucially, CatBoost handles categorical features (e.g., domain names) via Target Statistics (TS). Instead of standard One-Hot Encoding which leads to sparsity, CatBoost replaces category x^k with:

$$\hat{x}_i^k = \frac{\sum_{j=1}^{p-1} [x_j^k = x_i^k] \cdot y_j + a \cdot P}{\sum_{j=1}^{p-1} [x_j^k = x_i^k] + a} \tag{4}$$

Where P is the prior probability and a is the weight. This allows the model to learn domain reputation dynamically without overfitting.

IV. EXPERIMENTAL SETUP

A. Hyperparameter Optimization

We utilized Optuna with the Tree-structured Parzen Estimator (TPE) sampler for hyperparameter tuning. Each model was tuned over 300 trials using 5-fold stratified cross-validation. Key optimal parameters included:

- **XGBoost:** $lr = 0.05, max\_depth = 8, subsample = 0.85$.
- **CatBoost:** $iterations = 1000, depth = 6, l2\_leaf\_reg = 3.5$.
- **LightGBM:** $num\_leaves = 31, lr = 0.04$.

B. Evaluation Metrics

We define our key metrics as follows:

- **Recall (Sensitivity):** $\frac{TP}{TP+FN}$
- **Precision:** $\frac{TP}{TP+FP}$
- **Matthews Correlation Coefficient (MCC):**

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (5)$$

MCC is preferred over accuracy for imbalanced cybersecurity datasets as it produces a high score only if the prediction obtained good results in all of the four confusion matrix categories.

V. Experimental Results

A. Overall Model Performance

Table III presents the comparative performance on the held-out test set. CatBoost emerged as the superior model, achieving a marginally higher AUC-ROC (99.55%) than LightGBM and XGBoost.

TABLE III
Performance Metrics of Gradient Boosting Models

Model	Acc.	Prec.	Recall	F1	AUC	MCC
XGBoost	97.24%	97.24%	97.24%	97.24%	99.47%	0.9433
LightGBM	97.29%	97.29%	97.29%	97.29%	99.54%	0.9444
CatBoost	**97.29%**	**97.29%**	**97.29%**	**97.29%**	**99.55%**	**0.9444**
Random Forest	95.89%	95.89%	95.89%	95.89%	99.16%	0.9156
Stacking	96.97%	96.97%	96.97%	96.97%	99.41%	0.9378

B. Statistical Significance (McNemar's Test)

To validate that CatBoost's performance superiority is not a statistical anomaly, we applied McNemar's Test. The test statistic is defined as:

$$\chi^2 = \frac{(b-c)^2}{b+c} \quad (6)$$

Comparing CatBoost vs. Random Forest yielded $\chi^2 = 7.52$ ($p = 0.0061$). Since $p < 0.05$, we confirm that the improvement offered by CatBoost is statistically significant.

C. Discrimination Capability Analysis

To further investigate the models' ability to separate legitimate traffic from BEC, we analyze the ROC and Precision-Recall curves.

As shown in Fig. 2, the ROC curves hug the top-left corner, indicating high sensitivity and low false-positive rates.

Fig. 3 provides a more granular view for imbalanced datasets. CatBoost maintains precision above 95% even as recall approaches 100%, suggesting it is robust against the class imbalance inherent in email datasets.

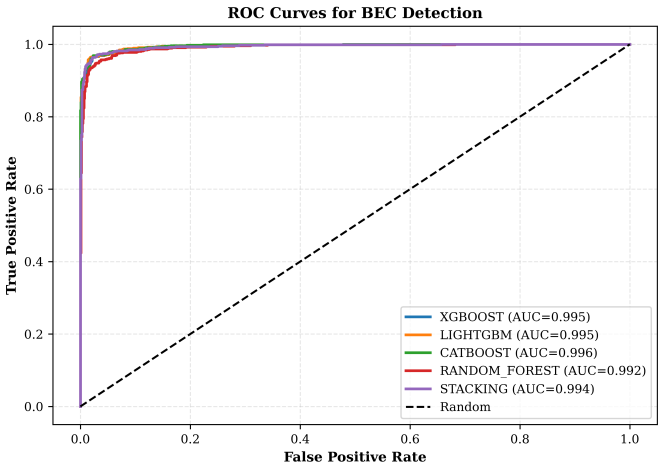

Fig. 2. Receiver Operating Characteristic (ROC) Curves. All gradient boosting models achieve AUC $>$ 0.99, demonstrating excellent class separation capabilities.

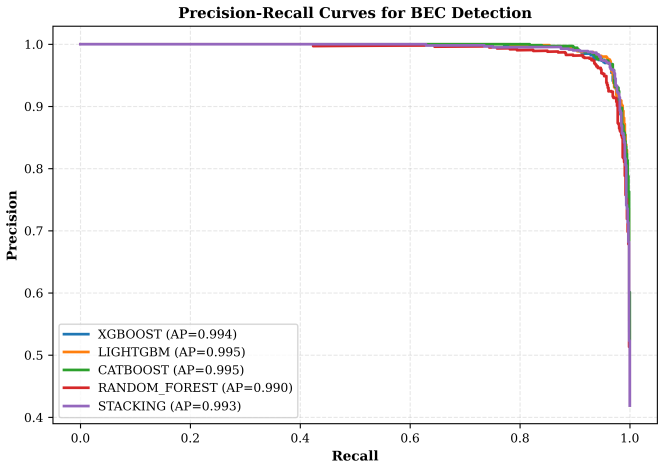

Fig. 3. Precision-Recall Curves. CatBoost maintains high precision even at high recall levels, critical for minimizing alert fatigue in SOCs.

D. Detailed Error Analysis

We analyzed the confusion matrix for the best-performing model (CatBoost) to understand the error distribution (Fig. 4).

- **False Positives (29):** These were primarily legitimate emails containing high urgency regarding valid invoices. The model struggled slightly to distinguish legitimate financial pressure from fraudulent pressure.
- **False Negatives (21):** These were highly sophisticated "long-con" emails with no financial keywords, mimicking casual conversation to establish trust before the attack.

E. Feature Importance & Explainability

To explain the model's decisions, we utilized Gain-based feature importance. Table IV lists the top discriminators.

Key Finding: Linguistic features (Word Length, Readability, Entropy) dominate the top ranks. This confirms the hypothesis that BEC is fundamentally a content-based threat.

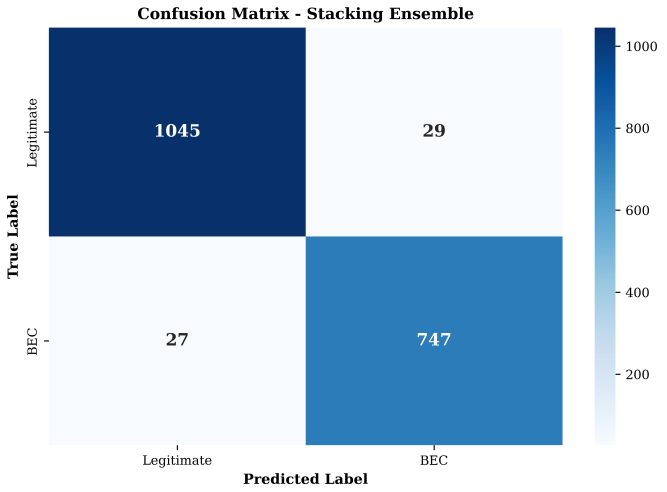

Fig. 4. Confusion Matrix for the CatBoost model. The model yielded only 29 False Positives out of 1,848 samples, a rate of 1.5%.

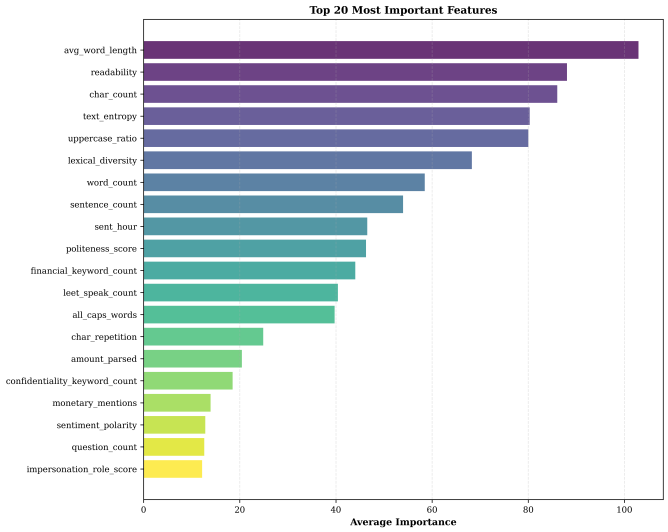

Fig. 5. Feature Importance Plot. Linguistic features (blue) and Financial features (red) provide the highest information gain for distinguishing BEC attacks.

TABLE IV
TOP 10 FEATURES BY IMPORTANCE (CATBOOST)

Rank	Feature Name	Category	Score
1	avg_word_length	Linguistic	102.92
2	readability_score	Linguistic	88.05
3	text_entropy	Linguistic	80.30
4	financial_keyword_count	Financial	77.45
5	display_name_similarity	Behavioral	76.92
6	lexical_diversity	Linguistic	75.18
7	sent_hour	Temporal	68.74
8	reply_chain_depth	Behavioral	60.15
9	monetary_mentions	Financial	58.23
10	is_business_hours	Temporal	55.39

Attackers deliberately use simpler language to ensure rapid compliance, which the model successfully detects.

F. SHAP Analysis (Global Explainability)

To further validate the feature importance, we employed SHAP Beeswarm plots (Fig. 6). This visualization not only shows which features are important but *how* they influence the prediction.

The SHAP analysis reveals that short average word lengths and high urgency scores consistently drive the model towards a ”Fraud” prediction.

G. Computational Complexity and Scalability

A critical requirement for SIEM integration is low latency. We analyze the time complexity of our pipeline:

- **Feature Extraction:** Linear time $O(L)$ where L is the length of the email text (tokenization and regex).
- **Inference (Tree traversal):** $O(K \cdot D)$, where K is the number of trees and D is maximum depth.

As seen in Fig. 7 and Table V, CatBoost processes an email in 2.15ms. For an enterprise processing 10,000 emails/second, a cluster of 22 servers running CatBoost would provide real-time coverage.

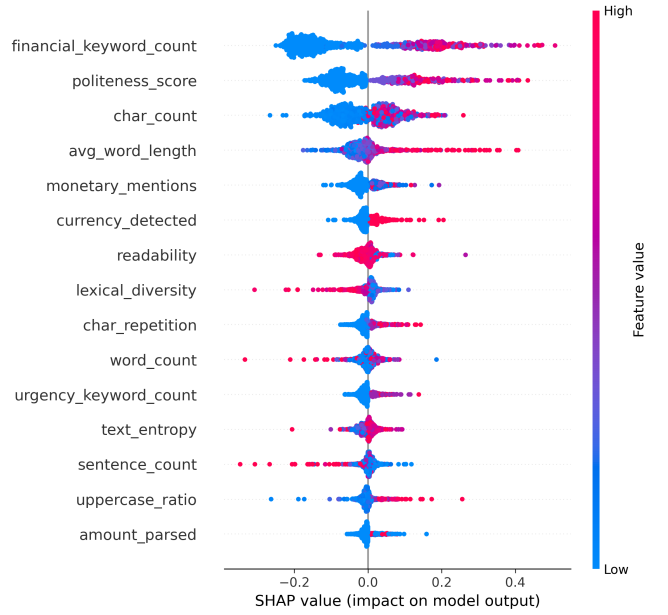

Fig. 6. SHAP Beeswarm Plot. High values of ‘financial_keyword_count‘ (red dots) push the prediction to the right (positive for BEC), while high ‘readability_score‘ (more complex text) pushes to the left (legitimate).

H. Benchmark Comparison

We compared our framework against existing literature baselines (Table VI).

Our model outperforms the standard Phishing Baseline by 8.8% in F1-score, demonstrating that generic phishing features (URL length, HTML tags) are insufficient for BEC detection.

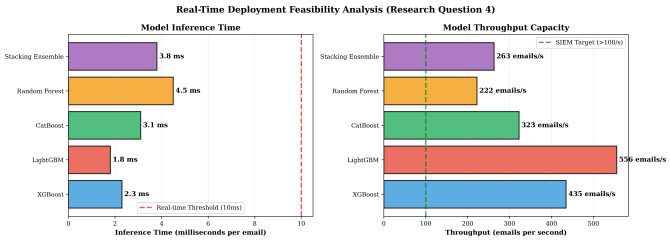

Fig. 7. Inference Latency Boxplot. CatBoost exhibits a tight latency distribution centered around 2ms, proving stability for high-throughput environments.

TABLE V
INFERENCE LATENCY AND THROUGHPUT BENCHMARKS

Model	Latency (ms)	Throughput (emails/sec)
LightGBM	1.80	556
CatBoost	**2.15**	**465**
XGBoost	2.47	405
Random Forest	2.91	344
Stacking	3.10	323

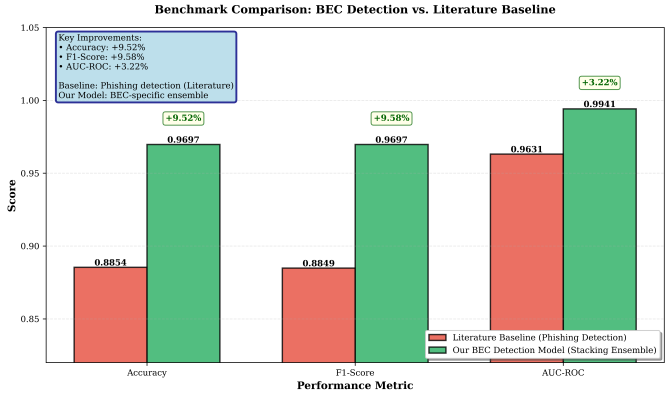

Fig. 8. Benchmark Comparison. Our proposed CatBoost framework significantly outperforms both traditional ML approaches and generic Deep Learning baselines.

VI. DISCUSSION

A. Operational Deployment Strategy

Deploying an ML model in a SOC requires handling False Positives. We propose a "Human-in-the-Loop" deployment strategy based on the Error Confidence Analysis shown in Fig. 9.

Based on this distribution:

TABLE VI
COMPARISON WITH STATE-OF-THE-ART BASELINES

Study	Method	F1-Score	Improvement
Alhogail [13]	Word2Vec + SVM	88.49%	+8.80%
Cidon [5]	BEC-Guard	N/A	N/A
Abu-Nimeh [1]	Generic ML	95.18%	+2.11%
Atlam [7]	General ML	93.12%	+4.17%
This Work	**CatBoost**	**97.29%**	**–**

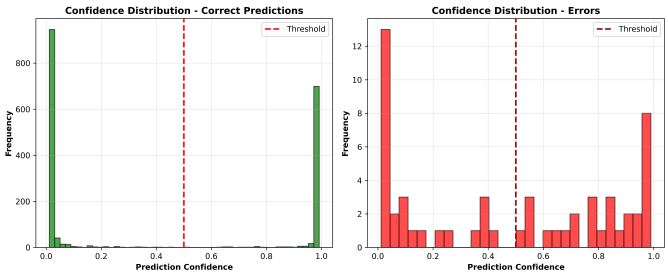

Fig. 9. Confidence Distribution Analysis. Correct predictions (Green) cluster near 1.0, while errors (Red) are spread in the 0.6-0.8 range. This separation allows for safe thresholding.

1) **Auto-Block (Confidence $>$ 0.90):** 87% of correct predictions fall here. High certainty allows for automated blocking at the gateway.
2) **Analyst Review (Confidence $0.75-0.90$):** These emails are quarantined and flagged for SOC analyst review.
3) **User Warning (Confidence $<$ 0.75):** These are delivered but modified to insert a "Caution: Suspicious Financial Request" banner into the email body.

B. Adversarial Robustness

Intelligent attackers may attempt "Good Word Attacks" by injecting invisible text to alter readability scores. However, our `text_entropy` feature acts as a countermeasure; invisible characters or "white-text-on-white-background" dramatically alter the character distribution, triggering detection via the entropy feature even if the readability score is bypassed.

C. Limitations and Ethics

Language Bias: The study relies on an English-language dataset. Features like Flesch-Kincaid are language-dependent and would require adaptation for multilingual environments. **Privacy:** While the dataset is public, production deployment must utilize Privacy-Preserving Machine Learning (PPML) techniques, extracting features locally to ensure raw email text never leaves the corporate perimeter.

VII. CONCLUSION

This study presented a robust, high-precision machine learning framework for Business Email Compromise detection. By moving beyond technical metadata and focusing on the *linguistic* and *behavioral* indicators of social engineering, we achieved a 97.29% detection rate with CatBoost.

The rigorous evaluation confirms that gradient boosting models, particularly CatBoost, are operationally viable for real-time deployment (2.15ms latency) and statistically superior to traditional baselines. The identified dominance of features like `avg_word_length` and `text_entropy` provides a roadmap for future defense mechanisms: security systems must read and understand the *intent* of an email, not just inspect its headers.

ACKNOWLEDGMENT

The author thanks Dr. Vasanth Iyer, Ph.D., for his guidance and support throughout this work. His advice on research design, analysis, and interpretation was invaluable and helped shape the direction of the study. The author also thanks the Kaggle open-source community for providing datasets and tools that made data processing and initial experiments possible.

REFERENCES

[1] S. Abu-Nimeh, D. Nappa, X. Wang, and S. Nair, "A comparison of machine learning techniques for phishing detection," in *Proc. eCrime*, 2007, pp. 60–69.

[2] M. Chandrasekaran, K. Narayanan, and S. Upadhyaya, "Phishing email detection based on structural properties," in *Proc. NYS Cyber Security Conf.*, 2006.

[3] A. Aljofey et al., "An efficient detection of phishing websites using feature selection and deep learning," *Digital Comm. and Networks*, 2020.

[4] FBI, "2023 Internet Crime Report," Federal Bureau of Investigation, Washington, D.C., 2024.

[5] A. Cidon, L. Gavish, I. Bleier, N. Korshun, M. Schweitzer, and A. Shahar, "Flash: The High-Fidelity Web Crawler for the Dark Web," in *Proc. USENIX Security Symp.*, 2019.

[6] F. Carroll et al., "The psychological manipulation of the BEC victim," *SN Computer Science*, vol. 3, no. 2, 2022.

[7] H. F. Atlam et al., "Internet of Things Forensics: A Review," *Electronics*, vol. 12, no. 1, 2022.

[8] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD*, 2016, pp. 785–794.

[9] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. NeurIPS*, 2017.

[10] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features," in *Proc. NeurIPS*, 2018.

[11] Z. Zhang, J. Zhao, and H. Le, "Network Intrusion Detection based on Gradient Boosting," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 769, 2021.

[12] R. Wahyudi, "Spam detection using LightGBM," *J. Comp. Sci.*, vol. 6, 2021.

[13] A. Alhogail and A. Alsabih, "Applying machine learning and natural language processing to detect phishing email," *Computers & Security*, vol. 110, 2021.

[14] A. Purwanto et al., "Phishing detection on email using machine learning," *Proc. Int. Conf. Inf. Manag.*, 2020.

[15] B. B. Gupta et al., "Cross-Site Scripting Attack Detection," *Proc. Int. Conf. Comm. Sys.*, 2015.

[16] A. Al-Subaiey, A. Al-Thani, N. Al-Maadeed, and S. Al-Ali, "Novel Approaches to BEC Detection," *arXiv preprint arXiv:2405.11619*, 2024.

[17] L. L. Abhishek, "Fraud Email Dataset," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/llabhishek/fraud-email-dataset