

Unsupervised Machine Learning Model for The Real-Time Prevention of Phishing

Gbolahan Babatunde Ajayi^{1,*}, Prof. E.J. Garba², Jibrin Baba Isah³, Abubakar Bello⁴

¹ Department of Computer Science, School of Information Technology and Computing, American University of Nigeria, Yola by-pass, Yola, Adamawa State, Nigeria.

² Department of Computer Science, Faculty of Computing, Modibbo Adama University, Yola, Adamawa State, Nigeria.

³ Department of Computer Science, Modibbo Adama University, Yola, Adamawa State, Nigeria.

⁴ Department of Information Technology, Faculty of Computing, Modibbo Adama University, Yola, Adamawa State, Nigeria.
Email: ¹gbolahan.ajayi@aun.edu.ng, ² e.j.garba@mau.edu.ng, ³jibrinbabaisah@mau.edu.ng, ⁴abubello92@mau.edu.ng

*Gbolahan Babatunde Ajayi

ORCID ID: <https://orcid.org/0009-0008-2667-5029>

Abstract—Every internet user is vulnerable to phishing or other forms of cyberattacks that, if care is not taken, using technology to store and secure soft assets would lose its essence since cybercriminals are devising new means of stealing these assets with minimum efforts. As of today, the world is struggling to overcome the menace of cybercrime. Phishing continues to be a crippling problem for many of today's Internet users as the attack continuously escalate in figures and sophistications. This study developed an unsupervised machine learning model that prevents real-time phishing in the financial sector. The methodology adopted in this study is a hybrid unsupervised ensemble methodology which combined Autoencoders, Generative Adversarial Networks, and Self-Organizing Maps to model benign URL pattern, detect and block/prevent phishing attempts as statistical anomalies. The dataset needed for this study were gathered from Openphish and UCI. Model from Openphish dataset had accuracy of 94.48%, it performed below the model from UCI dataset with accuracy of 97.24%. Ensemble model of ANN and DL is highly efficient in detecting and preventing phishing urls in real-time. More specifically, ensemble model of deep learning and artificial neural network is the future of anomaly detection that researchers can even use in novel approach. Hence it was suggested that future studies should apply the ensemble model used in this study for spamming detection and deepfake detection and prevention.

Keywords— *Phishing; Cyberspace; Cybercrime; Cybersecurity; Dataset; Denial-of-service attack; Man-in-the-middle attack; Ransomware.*

I. INTRODUCTION

The unlimited cravings of human being to have everything at the snap of his fingertips resulted in his quest for advanced technology which has enabled man to store almost all assets on the internet and able to access them without having to move from one location to another. Mankind forgot to plan for the lazy and unrepentant criminals in the world, hence, the menace of phishing which is a form of cybercrime. Cybercrime is the consequence of our high reliance on the

internet; these forms of crimes are illegal acts where the computer as a device is used either as a tool or a target.

Cybersecurity takes care of cyber threats and cyberattacks. [1] defined a cyber threat as any possible risk or harm to digital infrastructure, data, networks, or computer systems. These dangers cover a broad range of actions, motives, and players that could take advantage of weaknesses and jeopardize the safety of digital environments. Insider threats, hacking efforts, harmful software (malware), and even inadvertent human error are some of the many possible origins of cyberthreats. These dangers can vary from low-level ones like phishing emails to more complex and focused ones like nation-state actors' advanced persistent threats (APTs). According to [1], a cyber-attack, on the other hand, is a malevolent and intentional activity done with the intention of jeopardizing the availability, confidentiality, or integrity of digital assets. Threat actors, which might include private citizens, organized crime groups, hackers, or even government-sponsored organizations, plan cyberattacks. We have different types of cyberattacks, these include ransomware attacks that encrypt data for extortion, distributed denial of service (DDoS) assaults that overload a network with traffic to interfere with services, and data breaches that provide unwanted access to private data.

According to [2], the activity of defending computers, mobile devices, servers, electronic systems, networks, and data against malevolent attacks is the contemporary definition of cyber security. Other names for it include System Security, Information Security (INFOSEC), and Information Assurance (IA). It was entirely benign blunders that led to the creation of the first cyber malware infection. However, the flawless growth of the online cybercrime legal field has caused cybersecurity to advance quickly [3].

A phishing attack begins when a cybercriminal dispatches a deceptive message, frequently containing a link to a fraudulent domain that mimics a trusted website. Although these interfaces appear genuine to unsuspecting users, the attacker secretly monitors and manages them. The term



Received: 6-6-2026

Revised: 14-10-2026

Published: 31-12-2026

"phishing" originated in 1996, and the strategy has expanded drastically over the subsequent decades. While initially confined to email scams, modern phishing tactics leverage diverse communication vectors, including voice messages, SMS, instant messaging clients, interactive multiplayer video games, and social media platforms. The primary objective is to manipulate victims into navigating to these replica websites. Because these lookalikes mirror original systems so precisely, targets willingly input sensitive credentials, including usernames, passwords, or additional personally identifiable data [4].

These deceptive websites are engineered to illicitly harvest highly confidential data, including credit card numbers, personal identification numbers (PINs), social security numbers, banking credentials, and authentication passwords. Alternatively, cybercriminals utilize these malicious domains as a vector to deploy various forms of malware directly onto the victim's operating system. In some cases, ransomware is installed to shut the user off the system which will give the culprit enough time to extract necessary details from the systems once he gains access to the system. Sometimes, in the process the user is asked to pay ransom to retrieve the system which does not usually work; even if the user formats the system and gains access to the system without demand from the culprit, all the needed information has been extracted and thus the culprit can go ahead to utilize those details for his personal malicious gains [4].

Traditional mechanisms designed to mitigate phishing and broader cybercrimes have increasingly proven inadequate, fueling the pursuit of more robust alternatives like machine learning. To date, machine learning models have demonstrated significant efficacy in intercepting a wide array of cyber threats, including online scams, credit card fraud, email spam, and phishing operations. Consequently, advanced cybersecurity frameworks have become vital for blocking unauthorized system access. Historically, organizations relied heavily on static firewall security policies; however, these measures no longer offer sufficient defense against modern intrusions. This shortfall occurs because modern threat actors continuously engineer sophisticated evasion techniques that exploit operating system vulnerabilities and network communication configurations [5].

There is a machine learning technique to detect phishing that does not require labeled datasets, this offer fresher ways to combat financial crime, it is known as unsupervised machine learning [6]. This study therefore covers unsupervised machine learning for phishing detection and prevention of financial data both in real-time scenario and otherwise. Unsupervised learning is a pattern shift in phishing detection that allows algorithms to identify trends and anomalies in data without direct supervision. In contrast to supervised learning, which employs labeled instances of phishing for training, unsupervised learning methods like clustering identify phishing and other forms of cybercrime activities based on departures from expected behavior [6, 7]. By spotting unusual patterns or behaviors, phishing detection tools assist banks in preventing fraudulent transactions and minimizing financial losses and damage to their reputation. These technologies enable proactive phishing detection and reaction since they are more precise and capable than

traditional rule-based methods. Phishing detection solutions must be able to adapt to the dynamic and ever-changing techniques employed by cybercriminals in order to maintain the efficacy of phishing detection measures[6, 7]. This preserves consumer confidence and safeguards financial assets. To sum up, unsupervised phishing detection and prevention techniques present a ground-breaking approach to stopping financial crimes and enhancing banking cybersecurity. By utilizing the power of contemporary machine learning algorithms, banking institutions can efficiently identify and reduce unscrupulous activities, guaranteeing the safety and integrity of digital transactions [6]. Among the various machine learning techniques used in combating phishing are autoencoder (AE), generative adversarial network (GAN), and self-organizing map (SOM) which are the main techniques adopted in this research.

A. Statement of the problem

Every internet user is vulnerable to phishing or other forms of cyberattacks that, if care is not taken, using technology to store and secure soft assets would lose its essence since cybercriminals are devising new means of stealing these assets with minimum efforts. There is loss of revenue in cooperate entities in the financial industry such as the banks and private individuals using financial products. [8] argued that phishing incidents inflict profound financial damage on commercial enterprises, spanning the immediate expenditures of incident response, system remediation, and litigation to the long-term erosion of corporate revenue and consumer trust. Because the economic repercussions of a successful security breach can be catastrophic, robust cybersecurity protocols are vital to safeguard critical infrastructure sectors, including energy grid networks, transit systems, healthcare facilities, and financial institutions.

[8] further emphasized that the operational disruption and compromise of financial networks yield severe societal consequences, ranging from acute public safety hazards and systemic economic instability to the potential loss of human lives. To mitigate these catastrophic outcomes, machine learning has emerged as a premier proactive paradigm for fortifying modern cyberspace environments. Empirical data underscores the vulnerability of specific financial sectors to digital interference. [9] revealed that commercial banking organizations comprise an overwhelming 91% of targeted financial entities within the United States, whereas insurance corporations represent a modest 7%. Furthermore, their analysis of granular business divisions indicated that during the 2018 fiscal period, consumer retail operations and credit card service provisions constituted the most heavily impacted sectors, sustaining 39% and 25% of total recorded system intrusions, respectively. Global metrics emphasize that phishing is a borderless threat that transpires universally on a daily basis. [9] documented that international phishing campaigns escalated dramatically, recording 129.9 million individual incidents during the second quarter of 2019 alone. This volume represented a 21% surge compared to the corresponding fiscal quarter of 2018. Geographically, Greece sustained the highest concentration of localized attacks at 26.2%, followed sequentially by Venezuela, Brazil, Australia, and Portugal. When analyzing organizational

targets across these regions, commercial banking infrastructure faced the most severe saturation, receiving 30.7% of fraudulent emails. Meanwhile, online payment processing architectures accounted for 20.1%, global internet portals constituted 18%, and social networking platforms sustained 9% of the total distribution.

From observation, there can never be a single solution to phishing because there are many strategies these unscrupulous people employ in carrying out their nefarious activities and there are different aspects to it. "Identity theft" or specifically "phishing" is one of the deadliest security deficits of the users in the Internet. Phishing used to be perpetrated using electronic mails in the past, these days, it has escalated beyond that to the different forms of it that we have today.

The traditional supervised methods suffer from three critical limitations which stem from i) Label Dependency and Decay to ii) Zero-Day Vulnerability and iii) High False Negatives in Real-Time. Past researches are deficient of a robust, real-time framework that can operate without labeled data. That led to a pressing need for a hybrid unsupervised ensemble approach which establishes a "Gold Standard" of benign behavior and uses deep reconstruction (AE), adversarial density (GAN), and topological clustering (SOM) to detect deviations.

B. Aim and objectives of the study

The aim of this study is to develop an unsupervised ensemble machine learning model for preventing real-time phishing in the financial sector. The specific objectives are to:

- i. design a hybrid unsupervised ensemble architecture integrating deep reconstruction, adversarial density, and topological clustering;
- ii. develop a hybrid unsupervised ensemble model integrating Autoencoders (AE), Generative Adversarial Networks (GAN), and Self-Organizing Maps (SOM) for the real-time detection;
- iii. implement a real-time traffic interception and processing pipeline using FastAPI and MITM proxy configured with a Firefox browser proxy to facilitate live analysis of URL traffic and prevention of phishing attacks in financial sector; and
- iv. validate the model's performance against the 95th percentile threshold using the OpenPhish and UCI datasets as well as compare the model with performance of the models from authors reviewed in the literature.

C. Significance of the study

The study will not only help the society at large to guide against phishing but also be of immense benefits to the financial sector who would benefit a real-time prevention solution to phishing and solution to zero-day attack. Since the study focused on phishing which is of various forms, it established ways of stopping attacks on financial data which affects not just the financial institutions but the users of the services of the financial institutions such as payment platforms and platforms that involve payment using credit or debit cards, even internet banking and mobile banking. Users

of financial products are better informed on the phishing as a cybercrime and the financial sector has been duly advised on the need to educate users of their products. Experts in the field of computer science would also learn better ways of securing web applications from intrusions using models and algorithms. With the models from this study, phishing of various forms would be prevented and new forms of phishing would be detected and prevented as this study has come up with sophisticated and robust phishing prevention model.

D. Scope of the Study

The focus of this research work was to detect phishing urls and prevent them using ensemble model of Autoencoder (AE), Generative Adversarial Network (GAN) and Self-Organizing Map (SOM), especially in the financial sector. Phishing and other forms of cybercrimes are getting out of hands; hence we need drastic measures to stop the menace immediately. The research was necessitated by the need for quick solutions to phishing and all forms of cybercrimes in the world. We obtained datasets from two different repositories namely Openphish and UCI. Two deep learning algorithms and one artificial neural network algorithm would be ensembled for effectiveness. This model was trained on two datasets as previously mentioned: Openphish and UCI and their performances were evaluated in respect to the various models from the literature review aspect of this study.

II. LITERATURE REVIEW

This aspect of the study reviews relevant literatures to the study.

A. Adopted Theory

The adopted theories for this study are the cyber-attack theory and the anomaly detection theory. These theories are relevant to this study because the cyberattack theory believes that information is the central theme of any cyber-attack and it states that the success of cyber-attacks depends on the information owned by the attackers at the time of the attack and the information modified or gained during the attack. If the phisher does not have access to information from the victim (phish), there will not be phishing attack. Hence, the need for victims of phishing to be sensitive when online and not give out confidential information so as to stop phishing attack. On the other hand, anomaly detection theory believes that any financial transactions that deviates from normal behavior is classified as phishing.

B. Base models

Unsupervised ML base models employed in this study are discussed below.

Autoencoder

According to [10], autoencoder has a general architecture which includes an encoder, decoder, and bottleneck layer. According to [10], the basic Autoencoder is an auto-associative neural network that is derived from the multi-layer perceptron, which aims to replicate its input (i.e., the target output is the input). The basic AE consists of an input layer, a hidden layer, and an output layer. [10] further explained that an input layer, a hidden layer that forms the

encoding, and an output layer whose units match the input layer make up the general structure of a simple AE. This is equivalent to learning an approximation of the identity function because the outputs are identical to the input.

Generative Adversarial Network (GAN)

GAN is an unsupervised machine learning model that was introduced for estimating generative models based on an adversarial process. In the words of [11], as a significant development in artificial intelligence, Generative Adversarial Networks (GANs) provide a strong framework for creating simulated data that closely mimics real-world data. GANs, which are made up of two linked neural networks called the Generator and Discriminator, participate in a dynamic adversarial process that is changing the field of deep generative modeling. By coordinating this interaction, GANs push the boundaries of data generation in a variety of fields, from creating images to producing language, exhibiting a significant impact on changing how computers understand and reproduce complex data distributions [11].

This dynamic is made possible by the Generator (G) network, which is tasked with creating fresh data samples based on the input data distribution, and the Discriminator (D) network, which is focused on separating real data from fake data. A discriminator network assesses the veracity of the created data, whereas GAN, a generator network, generates the data. While the generator seeks to create data that is identical to genuine data, the discriminator learns to discern between real and fake data through an adversarial method [11].

Self-Organizing Maps

[12] explained self-organizing method as a competitive learning method; Self Organizing Maps (SOM) is the mostly used neural network technique. SOM is an unsupervised learning method that does not require the intervention of humans to learn and has no knowledge of the input data properties. Is explained by the inventor, self-organizing maps are biologically inspired neural network that encloses high dimensional vectors into a grid of neurons [12].

According to [12], SOM is classified as unsupervised machine learning model, the procedure of its application is to start by applying the typical algorithm to the inputs. once that is completed, it results in a grid of neurons that compete to represent each input. From each competition a winning neuron will emerge. Hence, if the neurons are labeled such that each one belongs to some class, then we can view the winner as the output label. Once that is done, we derive the typical classifier (input vector in; output label out). Even though there is no supervision in the training whatsoever, the post-training label is necessary in order to compute performance measures like accuracy [12].

C. EMPIRICAL REVIEW

[13] introduced a classification framework built around a Support Vector Machine (SVM) to identify deceptive online threats. Their system cross-references active user inputs against a centralized repository containing historical records of benign, questionable, and explicitly malicious requests. In assessing internal architectural variations, the researchers compared alternative kernel functions; their findings revealed

that the polynomial configuration achieved an accuracy rate of 84.5%, outperforming the radial basis function (RBF) kernel, which scored 82.6% [13]. Compared to conventional supervised architectures, the hybrid unsupervised ensemble model developed in their study drastically minimizes operational risk within the financial sector. While [13] reached an accuracy ceiling of 84.5% using a polynomial SVM kernel, their system leaves an unacceptable 15.5% vulnerability margin. In high-stakes banking environments, such an error rate translates to severe financial leakage from missed phishing campaigns or extensive operational disruptions due to falsely blocked legitimate customer transactions. This study resolves this performance bottleneck by introducing an unsupervised approach that bounds benign URL expectations at the 95th percentile, ultimately establishing a robust, real-time detection accuracy of 97.24% deployed inline via a FastAPI and mitmproxy framework.

[14] analyzed the growing security threats surrounding malicious web infrastructure, emphasizing the automated identification of fraudulent domains. By extracting features across multiple vectors including lexical domain configurations, web page content, SSL certificate attributes, and historical record metrics; the authors trained a series of supervised machine learning classifiers [14] Their empirical trials compared Decision Trees, Random Forests, Support Vector Machines, and Gradient Boosting implementations; among these, the Gradient Boosting architecture demonstrated superior diagnostic capability, reaching an accuracy ceiling of 92% [14]. [14] presented a multi-feature dataset evaluation, but their approach reveals critical vulnerabilities when compared against your high-performance banking defense architecture. A 92% accuracy ceiling means that 8% of web traffic is misclassified. In the financial sector, an 8% failure rate means thousands of banking customers are exposed to identity theft daily, and legitimate financial APIs risk being incorrectly blocked. Our model's 97.24% accuracy drastically mitigates this operational risk space. [14] relied on explicit feature extraction from domain names, online content, and SSL certificates. Attackers can bypass this easily by generating clean, valid SSL certificates (e.g., via let's Encrypt) and mimicking benign lexical structures. Because their tree-based models (Gradient Boosting, Random Forest) look for specific splits on these features, they fail against dynamic variations. Our Autoencoder (AE) maps raw URL structures into non-linear latent spaces without rigid feature reliance. Gradient Boosting constructs decision trees sequentially to minimize a loss function based on historical patterns. It cannot adapt to structural mutations intentionally designed by scammers to bypass security systems. Our GAN components introduce a discriminator that models the true data distribution of benign endpoints, allowing our system to identify stealthy, unseen structural deviations.

[15] developed a multi-class email classification system using deep artificial neural networks and Principal Component Analysis (PCA) to distinguish between SPAM, HAM, and phishing, achieving 95.32% accuracy. The approach combined UCI, CSDMC, and SpamAssassin datasets, extracting features such as HTML tags and URL structures for improved threat identification. While [15] achieved a notable 95.31% accuracy by transitioning to

multi-class deep learning, their email-centric architecture reveals significant structural and operational limitations when contrasted against your high-frequency financial defense framework. [15] relied on Principal Component Analysis (PCA) for feature selection. PCA is a linear transformation technique that struggles to capture complex, non-linear correlations within modern malicious URL parameters. Because phishing vectors in the financial sector change dynamically, linear reduction drops crucial behavioral markers. Our Autoencoder (AE) acts as a deep, non-linear bottleneck that compresses high-dimensional structural data without losing vital contextual connections. Their model evaluates semantic payloads within emails (HTML tags, JavaScript strings, body text). This approach provides zero protection if a user clicks a malicious link via an alternative channel like SMS (Smishing), WhatsApp, or scanning a QR code at a banking kiosk. By focusing directly on the network infrastructure level (URL string parameters and transit telemetry), our framework provides cross-channel protection regardless of how the victim encountered the link. An accuracy rating of 95.31% means nearly 5 out of every 100 encounters are misclassified. In a high-stakes banking environment, allowing 8% of zero-day attacks to slip through creates immense financial and legal liability.

[16] investigated a supervised deep learning framework designed to classify fraudulent online platforms. Utilizing a benchmark dataset obtained from the University of California, Irvine (UCI) machine learning repository, the researchers evaluated alternative neural network architectures [24]. Their empirical analysis demonstrated that a standard dense artificial neural network reached a classification accuracy of 94.8%, while a two-dimensional Convolutional Neural Network (Conv2D) configuration yielded a slightly lower accuracy ceiling of 93.6% [16]. Evaluating the supervised deep learning framework of [16] highlights core vulnerabilities when applied to high-stakes, real-time financial security. Siddiq et al. achieved a lower accuracy with their CNN (93.6%) than their standard neural network (94.8%). This drops because 2D Convolutional layers (Conv2D) are mathematically optimized for spatial grid data like images. Treating URL characters or text features as a 2D matrix without spatial coherence forces the network to look for non-existent vertical spatial dependencies, degrading performance. Our framework leverages an Autoencoder (AE) to extract non-linear latent vectors natively from sequential data, avoiding spatial distortion. In banking security, a 94.8% accuracy means a 5.2% systemic failure rate. Allowing 5 out of 100 phishing links to bypass the security wall exposes financial institutions to millions in fraud losses. By combining AE, GAN, and SOM engines into a collaborative unsupervised ensemble, your model minimizes this blind spot to achieve a robust 97.24% accuracy. Our GAN discriminator continuously models the core distribution of legitimate (benign) traffic instead, ensuring that any variation is detected as an anomaly regardless of its novelty. Siddiq et al. acknowledged in their own published analysis that their model requires extensive real-time feature extraction from URLs before classification can execute, creating a massive processing bottleneck. Our deployment architecture uses FastAPI and mitmproxy to intercept traffic mid-transit, feeding compressed latent

features directly to a fast Self-Organizing Map (SOM) grid for instantaneous, inline protection.

[17] evaluated machine learning models for phishing detection, finding that Logistic Regression achieved a peak accuracy of 83.83%, while Support Vector Machine and Random Forest reached 82.30% and 75.80% respectively. The study highlighted the limited efficacy of conventional supervised algorithms in identifying complex phishing threats, with the highest-performing model leaving a notable vulnerability gap. For more information, please consult the original research by [17]. Evaluating the supervised learning framework of [17] reveals profound structural and performance deficiencies when viewed from the perspective of high-security banking infrastructure. [17] relied on Logistic Regression as their best model, which peaks at a low 83.83% accuracy. In the financial sector, a model that misses roughly 16% of threats is an operational liability, exposing thousands of account holders to credential harvesting. Furthermore, their Random Forest model collapses to 75.80% accuracy, demonstrating that standard supervised tree architectures struggle with high-dimensional URL data. Our model's 97.24% accuracy compresses this risk space to a fraction of their baseline. Logistic Regression is a purely linear classifier that forces a straight decision boundary through data features. Because modern financial phishing attacks rely on intricate, multi-layered URL configurations, sub-domain masking, and stealthy redirection mechanics, linear models fail completely to recognize the underlying threat pattern. Our framework solves this by running an Autoencoder (AE) to extract complex, non-linear latent feature spaces natively. Supervised classifiers like SVM and Random Forest have zero capacity to simulate or defend against evolving threat landscapes. They can only make decisions based on fixed, historical data splits. By integrating a Generative Adversarial Network (GAN), our model benefits from an adversarial discriminator that learns the exact multi-dimensional boundary of benign banking traffic, making it highly sensitive to stealthy, zero-day variations that linear models miss entirely. Traditional feature selection and preprocessing methods require intensive manual engineering of text strings. This adds computational overhead and introduces human bias into what constitutes a threat feature. Our triple-engine ensemble uses Self-Organizing Maps (SOM) to map features topologically in an unsupervised manner, grouping traffic dynamically based on mathematical proximity rather than rigid, pre-defined labels.

[18] compared machine learning and deep learning models for phishing detection using the UCI dataset, finding that a 1D-CNN achieved 96.75% accuracy without feature selection, outperforming K-NN (94.34%) and LSTM (91.23%). The study utilized feature selection to enhance performance and reduce execution times for the classifiers. Evaluating the machine learning and deep learning framework of [18] provides a valuable point of comparison, as their 1D-CNN achieves a strong 96.75% accuracy. However, looking at this through the lens of a high-security financial defense system reveals key architectural and operational limitations. [18]. found that their 1D-CNN (96.75%) outperformed their LSTM model (91.23%). While 1D-CNNs excel at extracting local, stationary n-gram patterns from text strings, they have a limited receptive field

and cannot capture long-range, non-linear contextual dependencies across an entire URL structure. Their LSTM struggled because standard recurrent networks fail with very long sequential structures due to vanishing gradients. Our Autoencoder (AE) resolves this bottleneck by compressing high-dimensional text vectors into a dense, global latent space that preserves complex structural relationships without positional loss. Their models were trained using a highly curated, relatively balanced subset of the UCI repository (4,898 phishing vs. 6,157 legitimate websites). Real-world financial network streams are highly imbalanced, where benign requests outnumber attacks by orders of magnitude. Supervised classifiers trained on artificial 50/50 splits suffer from high False Positive Rates when deployed in live environments. By using an unsupervised ensemble that models the true probability distribution of normal traffic, our framework adapts naturally to live, heavily skewed operational data streams. 1D-CNNs are notoriously vulnerable to subtle adversarial manipulation. An attacker can bypass a supervised CNN by inserting harmless sub-domains or padding the URL string with benign keywords, altering the local character combinations the filters look for. Our model incorporates a Generative Adversarial Network (GAN) discriminator trained to understand the global structural limits of benign financial traffic, making it much harder for sophisticated, obfuscated zero-day links to slip through. While 96.75% is a strong score for standard applications, the remaining 3.25% error margin is highly risky for enterprise banking platforms. Our AE + GAN + SOM ensemble achieves 97.24% accuracy by combining three deep engines, minimizing the individual blind spots that limit single-model architectures.

[19] constructed and benchmarked four computational variations to assess the performance of machine learning algorithms in isolating fraudulent web domains. Using the standardized UCI phishing domains dataset as a performance baseline, the authors engineered classification systems utilizing artificial neural networks (ANNs), support vector machines (SVMs), decision trees (DTs), and random forests (RFs) [19]. Their comparative metrics indicated that the random forest configuration achieved the highest classification capacity at 97.3% accuracy, outperforming its internal counterparts, DT at 96.3%, ANN at 95.5%, and SVM at 94.66%, while matching or exceeding historical baselines in the literature [19]. [19] present a strong supervised baseline, with their Random Forest model achieving a competitive 97.3% accuracy. However, a deeper look through the lens of a real-time banking security architecture reveals structural limitations that your unsupervised ensemble directly resolves. While their Random Forest model hits 97.3% accuracy, this score is highly dependent on the static feature distributions within the historical UCI dataset. Tree-based ensembles function by making crisp, orthogonal splits on explicit parameters (e.g., URL length, presence of prefixes). Phishing operations in the financial sector easily bypass these hard boundaries by purchasing clean, short domains that mimic banking sub-domains. Because our model uses an Autoencoder (AE), it converts raw strings into dense, non-linear latent spaces that map semantic intent rather than rigid, easily spoofed string characteristics. Random Forest models look backward; they

can only evaluate current data against a fixed history of tree splits. They possess no built-in mechanism to anticipate or model structural adaptations. By combining our AE with a Generative Adversarial Network (GAN), our framework actively trains against structural boundaries using a discriminator that learns the complete data distribution of benign traffic. This gives your architecture higher resilience against zero-day mutations designed specifically to fool supervised classifiers. Although 97.3% accuracy appears statistically similar to your 97.24% accuracy, the operational mechanics under the hood are completely different. A supervised Random Forest cannot control its False Positive Rate (FPR) deterministically; it optimizes for overall accuracy, which often leads to blocking legitimate traffic in a real-world environment. Our system handles this trade-off by locking the detection boundary at a strict 95th percentile threshold for benign URLs using a Self-Organizing Map (SOM). This mathematically guarantees a capped false alarm rate, which is an operational necessity for smooth banking transactions. Random Forest models require complete feature extraction across a large set of decision trees before making a final classification call. This creates a significant computational footprint when scaling up to thousands of concurrent banking user sessions. Our framework uses FastAPI alongside an inline mitmproxy architecture on Firefox, which intercepts and routes compressed latent variables instantly mid-transit, blocking malicious requests before the browser can even render the script.

Research by [20] examined the technical efficiency of machine learning architectures in classifying malicious and abusive web infrastructure. Their methodology evaluated structural and historical datasets of phishing and benign URLs, focusing on explanatory features such as cryptographic HTTPS presence, total URL string character count, and domain registration lifecycles [20]. When training traditional supervised benchmarks including Support Vector Machines, Decision Trees, and Random Forests, empirical outcomes demonstrated that the Random Forest model secured the highest overall classification capacity at 97% accuracy [20]. Concurrently, the Support Vector Machine exhibited robust generalization, achieving precision and recall metrics of 0.92 and 0.95, respectively [20]. While [20] present an insightful examination of URL metadata, their framework exhibits several core vulnerabilities when evaluated from the perspective of a real-time banking security architecture: [20] heavily weigh external attributes like domain registration dates and HTTPS presence. Modern financial phishers manipulate these metrics with ease; they frequently compromise aged, legitimate third-party business websites or instantly spin up valid SSL/TLS certificates via automated, free certificate authorities. Because their Random Forest and SVM models rely on explicit tabular thresholds for these features, the system is highly prone to evasion. Our Autoencoder (AE) captures deep, non-linear structural sequences of raw URLs natively, rendering the model resilient to superficial metadata spoofing. Their highest generalization metric for catching attacks is the SVM's 0.95 recall. In an enterprise financial deployment, a 0.95 recall means a dangerous 5% of active phishing attacks slip through undetected. Allowing 5 out of 100 fraudulent banking clones to bypass a security perimeter presents massive liability and

asset loss risks. Our AE + GAN + SOM ensemble eliminates individual model blind spots, elevating overall predictive performance to a secure 97.24% accuracy. Supervised classifiers require completely balanced, pre-labeled sets of historical attacks to find separating hyperplanes. However, real-time banking network streams are heavily imbalanced, composed of millions of benign queries and rare, novel attacks. Supervised models struggle with real-world distribution shifts, leading to spikes in false alarms. Our model uses a GAN discriminator to map the true mathematical distribution of benign traffic in an unsupervised fashion, allowing it to isolate anomalous transactions purely through deviation. Extracting features like "domain registration date" requires executing active WHOIS database lookups or calling external security APIs during a user's web session. This dependency introduces extensive processing latency that disrupts real-time web operations. Our deployment relies on FastAPI and mitmproxy, which process routing inputs instantly mid-transit, executing threat-scoring maps natively on a Self-Organizing Map (SOM) grid before browser rendering.

[21] introduced a heuristic-based machine learning framework designed to detect multi-channel phishing attacks across URLs, emails, and websites. To evaluate URL-based threats, the authors extracted 56 distinctive features, yielding a detection accuracy of 97.2%. Their email phishing detection component achieved an accuracy of 97.4%. Furthermore, by optimizing the feature space to 48 critical attributes for website phishing analysis, their model attained its highest performance tier with a 98.1% accuracy rate. [21] developed a multi-vector phishing detection system targeting URLs (56 features, 97.2% accuracy), emails (97.4% accuracy), and websites (48 features, 98.1% accuracy) using heuristic-driven machine learning classifiers. Both systems achieve highly competitive accuracy profiles in the 97% to 98% tier. Both methodologies recognize that diverse phishing vectors (URLs/traffic metadata) require optimized, multi-dimensional feature spaces to maximize detection rates. Jayaprakash *et al.* utilize a supervised, heuristic-dependent paradigm requiring extensive feature engineering (up to 56 features) from both benign and malicious classes. In sharp contrast, our model uses an unsupervised ensemble (AE+GAN+SOM) that models *only* the benign data distribution, completely removing the dependency on labeled phishing samples. The primary limitation of [21].’s work is its heavy reliance on static heuristics. Attackers can easily bypass these features using zero-day obfuscation techniques. Additionally, computing 56 heuristic features introduces severe processing overhead, creating a bottleneck for the real-time constraints required in financial monitoring. This study serves as a baseline showing the upper performance limits of traditional supervised heuristics. Our research directly addresses its gaps by showing how an unsupervised ensemble (AE+GAN+SOM) bound by a strict 95th percentile benign threshold can deliver comparable accuracy (97.24%) while solving the real-time processing and zero-day vulnerabilities inherent in the Jayaprakash *et al.* framework.

[22] conducted a comparative study of various machine learning algorithms to identify deceptive websites using key URL significance features. Their objective was to determine and deploy the most effective model for website phishing

identification. The researchers benchmarked three specific techniques: Logistic Regression, Multinomial Naive Bayes, and XGBoost. Ultimately, the Logistic Regression model surpassed the other classifiers, delivering the highest overall system performance with a peak accuracy of 96.63%. [22] executed a comparative evaluation of traditional machine learning classifiers, demonstrating that a simple Logistic Regression model outperforms Multinomial Naive Bayes and XGBoost by achieving an accuracy of 96.63% using URL-specific indicators. Both [22] work and our research target website-level phishing threats by focusing on statistical boundaries. Both studies prove that structured data engineering can yield highly competitive accuracy scores in the 96% to 97% range. The foundational difference lies in the model complexity and learning paradigm. Neeraja *et al.* relied on shallow, supervised classifiers (specifically linear and probabilistic models) that necessitate labeled training sets of both phishing and authentic URLs. Conversely, our framework uses a deep, unsupervised ensemble (AE+GAN+SOM) that learns complex, non-linear representations of benign data without ever needing exposure to malicious samples during training. A major flaw in [22] findings is that Logistic Regression, a purely linear classifier won the comparison. This strongly implies that their dataset was overly simplistic or linearly separable, which does not accurately reflect highly sophisticated, modern financial phishing campaigns. Furthermore, their reliance on static URL features means their model can be completely blindsided by active cloaking, reverse proxies, and dynamic domain generation algorithms (DGAs). This paper underscores the performance limits of standard, off-the-shelf supervised classifiers on basic URL datasets. Our research builds directly upon this limitation; where Neeraja *et al.* reach a ceiling of 96.63% using linear assumptions, our AE+GAN+SOM ensemble achieves a superior 97.24% accuracy by mapping complex, multi-dimensional anomalies at a strict 95th percentile benign threshold.

[23] developed a phishing classification framework that utilizes tree-based machine learning structures, with focus on Random Forest and Decision Tree models for evaluating web links. The system successfully isolated deceptive URLs from authentic ones, yielding an accuracy rate of 87.0% when applying the Random Forest ensemble and a lower accuracy of 82.4% with the standalone Decision Tree classifier [23] present a supervised framework employing tree-based architectures specifically Random Forest and Decision Tree classifiers to separate legitimate URLs from malicious threats, achieving detection accuracy rates of 87.0% and 82.4%, respectively. Both [23] work and our research target website-level phishing threats by attempting to map and isolate malicious infrastructure. Both studies recognize that feature patterns within web traffic can be categorized to prevent users from falling victim to identity theft. The structural contrast is stark. Mandadi *et al.* depend entirely on supervised tree classifiers that split data based on explicitly labeled features of both benign and phishing data. In contrast, your model integrates a deep, unsupervised ensemble (AE+GAN+SOM) that models the latent distribution of *only* benign data, completely removing the requirement for malicious training data. The primary limitation of [23].’s approach is its weak accuracy ceiling (peaking at 87.0%),

which leaves a catastrophic 13% vulnerability gap completely unacceptable for the zero-tolerance risk profile of the financial sector. Furthermore, traditional decision trees are notorious for overfitting on static training data and fail completely when confronted with dynamic, zero-day phishing links that modify text patterns. This paper establishes a clear empirical justification for our research. By highlighting the unreliability and low accuracy of shallow supervised trees (82.4%–87.0%), it directly supports our shift toward a deep, unsupervised anomaly-detection approach (AE+GAN+SOM) with a strict 95th percentile threshold to eliminate dangerous false negatives.

[24] engineered a multi-attribute threat detection system aimed at identifying malicious URLs by extracting robust features from deceptive webpages and infrastructure hosts. To resist adversarial spoofing, their framework focused on indicators that are hard for cybercriminals to alter. The researchers initially applied traditional machine learning classifiers, achieving low accuracies of 64% using Support Vector Machines (SVM) and 67% using Random Forests (RF) for email threat detection. To address these performance deficiencies and enhance resilience against massive phishing campaigns, the authors transitioned to a Deep Learning paradigm, using a Convolutional Neural Network (CNN) that raised the peak detection accuracy to 70%. [24] introduce a malicious URL and email classification system using host-based and webpage features, benchmarking traditional classifiers like SVM (64% accuracy) and Random Forest (67% accuracy) against a deeper Convolutional Neural Network (CNN) that yields an accuracy ceiling of 70%. Both systems recognize the necessity of transitioning from traditional, shallow machine learning algorithms (such as Random Forest) to sophisticated, multi-layered neural network architectures to capture complex, hidden patterns within malicious internet data. The core structural variance rests on data dependencies and learning paradigms. [24] deployed a supervised CNN that requires explicit labels for both benign and malicious data classes, capping their performance at a low 70%. Our model relies on a deep, unsupervised ensemble (AE+GAN+SOM) that trains entirely on clean, benign baseline data to achieve an accuracy of 97.24%. A critical limitation of Vargheese and Sreelakshmi's system is its remarkably low accuracy profile (70% peak), which leaves an unacceptable 30% breakthrough rate for phishing threats. This fatal gap makes it highly dangerous for real-time deployment in high-stakes financial sector transactions. Additionally, supervised CNNs often suffer from high inference latency and fail to generalize when face-to-face with zero-day structural variations. This paper underscores the inherent performance bottlenecks faced by supervised neural networks when trying to map diverse, adversarial phishing distributions. Our study directly resolves this constraint by demonstrating that switching to an unsupervised anomaly detection paradigm with an optimized 95th percentile threshold yields a vastly superior accuracy rate (97.24%) while minimizing false negatives.

[25] introduced a multi-tier classification framework designed to identify deceptive webpages by analyzing a combination of URL syntax, textual information, and visual content. Utilizing a substantial collection of 20,000 web links, the researchers extracted 22 distinct URL-based

attributes while simultaneously building an independent text dataset for Natural Language Processing (NLP) assessment. To counteract advanced tactics where phishing indicators are masked inside graphics, the system integrated optical text extraction from website images to determine legitimacy. The framework benchmarked eight distinct classifiers, including Support Vector Machines (SVM), Random Forest, and Multilayer Perceptron (MLP). Empirical testing demonstrated that XGBoost achieved the top performance tier, delivering 94% accuracy during training and 91% during testing. The MLP and Random Forest classifiers both matched this 91% testing accuracy, closely followed by the Decision Tree at 90%. For standalone text-based evaluation, Logistic Regression and SVM produced accuracies of 87% and 88%, respectively. [25] presented a multi-modal, layered classification framework using 22 structural URL features alongside NLP-driven text and image-text extraction, benchmarking eight supervised algorithms to achieve a peak testing accuracy of 91% using XGBoost. Both frameworks acknowledged that phishing detection requires a multi-dimensional view of data features. Both approaches use complex ensemble methods (XGBoost vs. our AE+GAN+SOM) to evaluate threat indicators and protect end-users from malicious infrastructure. [25] construct a heavy, multi-modal supervised model that relies on complex feature engineering across three domains (URL, NLP text, and image text extraction). This requires massive, balanced datasets of labeled phishing examples. In direct contrast, our model operates on a deep, unsupervised paradigm that bypasses multi-modal extraction entirely by training exclusively on pure benign transaction distributions. A major operational flaw in [25]'s model is its high computational complexity. Parsing URL features, processing NLP text, and running optical text extraction on images simultaneously creates severe processing delays. This makes it impossible to use for real-time financial streams

III. RESEARCH METHODOLOGY

This aspect of the study discussed the design and development aspects of the methodology, with the methodology being clearly spelt out. The methodology has two aspects, which include the design and development; the design aspect has to do with the conceptual framework and the theoretical framework.

A. Adopted Technique

Since phishing is anomalous in nature, due to the irregular pattern of the phishing URLs compared to authentic URLs, the study adopted a hybrid unsupervised anomaly-detection ensemble model which combines deep reconstruction, adversarial density estimation, and expert rule-based signals, with score-level fusion and calibrated thresholding. Phishing is anomalous in nature because phishing URLs systematically deviate from the statistical patterns of legitimate URLs /links. The study did not rely on a single signal; rather, it combines an autoencoder (reconstruction anomaly score), a generative adversarial network (likelihood/discriminator confidence) and a self-organizing map (cluster distance/BMU deviation); all these are aggregated into a composite anomaly score. Self-

organizing map is artificial neural network algorithm while both AE and GAN are modern deep learning algorithm which are suitable for phishing detection. According to [26], deep learning technology originated from artificial neural networks (ANNs). Some researchers who have made use of ensemble and expert rule-based methodology are reviewed next.

B. Theoretical Framework

This aspect is the design stated. In robustness, if a sophisticated phishing URL mimics the visual structure of a bank (tricking the AE's reconstruction), the GAN-based logic may still identify it as a synthetic outlier, or the SOM might place it in a topologically suspicious "neighborhood". When it comes to real-time performance, our base must prioritize low-latency inference. iii) Financial cybersecurity requirements. Our research base is specifically tailored to the high-stakes nature of finance, where the cost of a "False Positive" (blocking a real customer) is nearly as high as a "False Negative" (allowing a phishing attack). The unsupervised nature of our base aligns with data privacy regulations (like GDPR), as these models can often be trained on local, decentralized data without exposing raw sensitive transaction details.

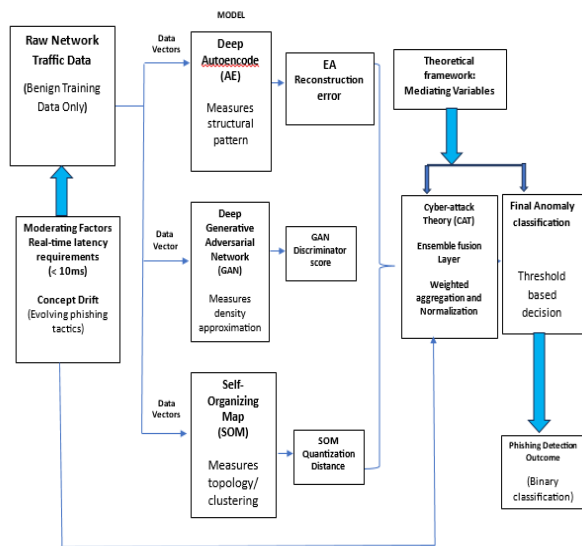


Fig. 1. Theoretical Framework for the Hybrid Unsupervised Framework for the Current Study

Fig. 1 shows the diagrammatic illustration of the theoretical framework. It shows the independent variables to comprise of the, URL Features, Domain Attributes, Content, Behavioral Metrics. It further illustrates the mediating variables, or mediators, which are variables that intervene and explain the basic process of how the independent variables and the dependent variables relate. This mediator acts as a bridge; the independent variable initiates the mediator; this prompts the dependent variable. The mediating variables are the AE reconstruction error, GAN discriminator score, and the SOM quantization distance.

The independent variables (known as predictors): These are structural inputs (such as URL Features, Domain Attributes) and contextual inputs (such as Content, Behavioral Metrics) form the basis for analysis. The

mediating variables (Internal Mechanisms): These bridge the input features to the final outcome. The system calculates an anomaly probability via AE Reconstruction Error, GAN Discriminator Score, and SOM Quantization Distance. The dependent variable (used in this research as the effect): is the phishing detection outcome (Anomaly vs. Legitimate) is the final state triggered when the integrated score crosses the calibrated 95th percentile threshold adopted in this study. For integration & aggregation strategy score normalization: Each mediating variable operates on a different scale; the framework applies normalization to ensure all three signals contribute equitably. As for the weighted aggregation: The final anomaly index is a weighted sum of these normalized signals, allowing for the prioritization of specific models (e.g., favoring the AE for structural anomalies). When it comes to moderating factors (i.e. external constraints), the relationship between my inputs and the detection outcome is moderated by three critical factors namely: i) Regulatory Compliance: This limits the type of data stored or analyzed (e.g., PII masking). ii) Real-Time Latency: The framework requires low-latency execution to ensure the Firefox + mitmproxy setup does not degrade user experience.

The score normalization & ensemble fusion which i-s the process scores are adjusted from different scales, distributions or sources to a comparable standard, such as a 0-1 range or a standard normal distribution as used in this study [0, 1]. This score normalization is essential in order to have fair comparison across tests, datasets, or models thereby preventing features with large scales from dominating; if features from large scale dominate, it will have a negative effect on the model. This study prevented that from taking place. Next comes the weighted aggregation of anomaly scores, which can be seen as a moderating variable; this is a third variable that changes the potency or trend of the association between the cause, which is the independent variable, and the effect, which is the dependent variable.

To summarize the theoretical framework, the theoretical framework is based on cyber-attack theory, which indicates that attack success depends on info gain/modification, and its implementation components are feature engineering and adversarial modeling. The anomaly aspect is the classification of phishing as anomaly because it deviates from normal transaction URLs' behaviour and it depends on malicious acts, which are statistical outliers while its implementation components are AE Reconstruction & SOM Topological Mapping. The last aspect, as revealed in Fig. 1, is the ensemble aspect, which combines weak signals to create strong defense, and its implementation components are Score-Level Fusion & Calibrated Thresholds.

C. Conceptual Framework

This is the second aspect of the design in chapter three as stated also in the study's objectives. The sole aim of the conceptual framework in the thesis is to provide a well-structured outline of the key concepts, components, relationships, and processes involved in addressing a problem. For this study, it is an organized summary of the main ideas, elements, connections, and procedures involved in solving phishing as a challenge facing the financial industry and all internet users who use financial products

from the financial industry. This describes how financial systems, like email gateways, online transactions, and banking apps, can use unsupervised machine learning (ML) to detect and prevent phishing assaults in real time. Phishing in the financial industry often uses deceptive URLs that impersonate trustworthy banks or other financial institutes in order to gain login credentials or start fraudulent transactions. Because it doesn't require labeled data, which is hard to get by for emerging threats, and because it concentrates on anomaly detection, finding deviation from typical patterns, such odd login attempts or dubious URL structures; this study's adopted unsupervised machine model is a perfect choice. For this study, the framework is structured into five major concepts.

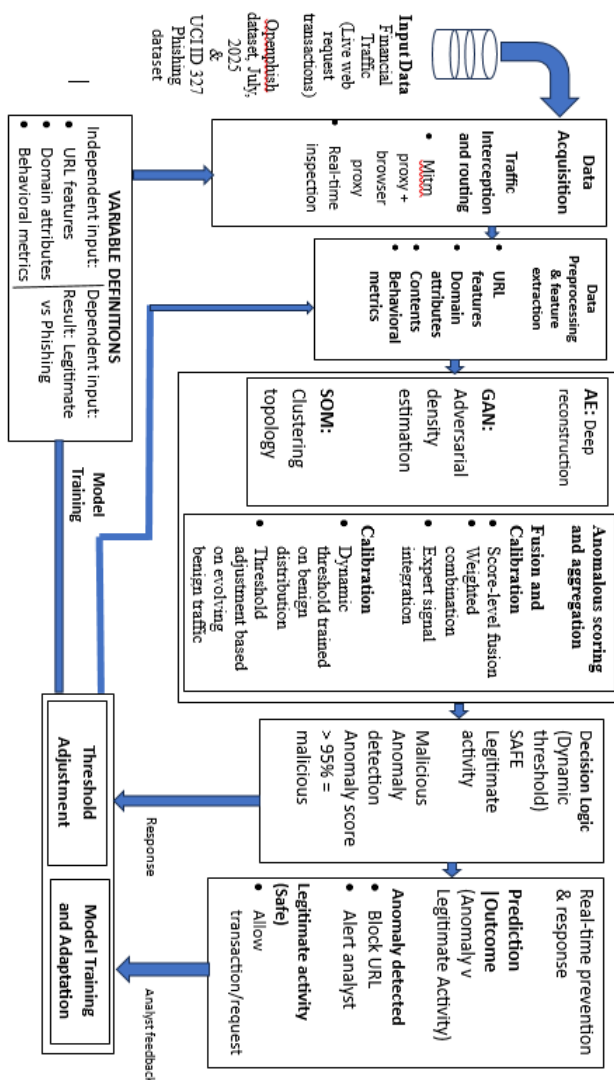


Fig. 2. Conceptual Framework: Unsupervised Machine Learning Model for Real-Time prevention of Phishing in the Financial Sector (current study)

Fig. 2 shows the diagrammatic illustration of the conceptual framework. The first component is the input sources in the financial sector, and they include the URLs, Email metadata, transaction logs, domain registration data, website structural features, as well as user behavioral

patterns. The second is the feature engineering aspect that transforms raw data into URL lexical features (such as length, special characters, entropy), domain-based features (age, DNS records), as seen among the features for both the OpenPhish and UCI datasets. Included are the content-based features, that is, HTML structure, and then the behavioral features (click timing, session duration), these are not part of the features in the datasets collected for the hybrid ensemble model of this study. The statistical normalization (Min-Max scaling to $[0,1]$) as seen in the architecture of the model developed as well as the unifying equation for the study.

The independent variables (Inputs) are the categorical features that serve as the foundation of your detection system and these are: i) URL Features: Structural indicators (e.g., URL length, dots in domain). ii) Domain Attributes: Technical metadata (e.g., DNS age, SSL status), iii) Content: Page-level data (e.g., HTML structure behavior). iii) Behavioral Metrics: User-session dynamics intercepted via mitmproxy.

The mediating ensemble & score fusion is the relationship defined by how the inputs are transformed into a risk signal. When it comes to transformation, the independent variables are fed simultaneously into the AE (Deep Reconstruction), GAN (Adversarial Density), and Expert Rules. The score-level fusion, here each module outputs a partial risk score. These are then merged through Score-level fusion to create a unified integrated anomaly index. The dependent variable (which is output) is the final outcome. Phishing detection is determined by comparing the integrated anomaly index against the model's calibrated threshold. Determining legitimate activity is done by verifying if the score is $\leq$ the 95th percentile; then the system confirms a "Benign" state. For Phishing / Anomaly, if the score $\geq$ the 95th percentile, the system triggers a "Malicious" outcome.

A hybrid ensemble mode from the unsupervised learning used in this study included an autoencoder (AE) for reconstruction error detection; a generative adversarial network (GAN) for adversarial anomaly scoring; and a self-organizing map (SOM), topological clustering & distance mapping. These models produce independent anomaly scores. The score fusion & threshold mechanism, score normalization scale to $[0,1]$, that is min – max scaling as explained previously under the theoretical framework. Hence, the weighted ensemble

combination is explained in equation 3.1 of the mathematical model. For features with large scales not to dominate the model, which may have a negative impact on the model, the study employed min – max scaling. In this hybrid ensemble model, scaling to a $[0, 1]$ range is regarded as the "great equalizer." Each model measures "weirdness" differently; in fact, the imbalanced nature of the OpenPhish dataset would have negatively impacted the model, but scaling ensured that a small reconstruction error from the AE isn't ignored just because SOM outputs large distance numbers. The model was trained on the 95th percentile of the benign distribution, which is the adaptive threshold for real-time detection.

Result of the scaling: Score-Level Fusion

The study did not use hybrid separately or fusion separately; as explained in the methodology, the study adopted a hybrid unsupervised ensemble anomaly-detection model which combines deep reconstruction, adversarial density estimation, and expert rule-based signals, with score-level fusion and calibrated thresholding. Once scaled, the weighted aggregation is given as Final Score = $S_{final} = w_1 S_{AE} + w_2 S_{GAN} + w_3 S_{SOM}$ which is the unified mathematical model for the three and the adding of this three is equal to 1 as revealed in the architecture of developed model. Because every input is now between 0 and 1, the 95th percentile threshold can be set accurately across the entire ensemble.

D. Model Design

To fulfill the first objective of this study, this aspect takes care of the design in which the blue-print for the model to be developed is established. This design is explicitly detailed above; it took care of theoretical framework and the conceptual framework respectively. The methodology is divided into system design and development. For the model in this study, the design aspect is about the logic, the math, the conceptual and theoretical framework as illustrated in figure 3.1. The mathematical model is clearly stated in the section here. The structural design (the blueprint) has to do with why the study chose the three models ensembled which is to capture the latent "benign" manifold of financial URLs; the ensemble topology which shows that the model adopts parallel processing via FastAPI with score-level fusion as indicated in the methodology. Mathematically/logically the scores merge through the fusion formular given as the final design: $S_{final} = w_1 S_{AE} + w_2 S_{GAN} + w_3 S_{SOM}$. The strategy was to us scaling strategy by using Min-Max scaling to the [0, 1] range. When it comes to the process design, that is, the flow, data interception is done by designing the handshake between Firefox, mitmproxy, and the FastAPI endpoint. The design will be incomplete without mentioning the thresholding which has to do with designing the calibration phase where the model "sees" only benign data to set that 95th percentile mark, anything greater than the 95th percentile mark is regarded as phishing.

In summary, the study designed the model through system architecture diagrams and algorithms. The architectural diagrams are theoretical framework, conceptual framework and the proposed model architecture which later became the developed model architecture in figure 3.4. The logic flow is displayed in all these three figures, especially in figure 3.4. The algorithms/pseudocode depicting the design of the algorithm are shown in the next sections.

E. Dataset Collection and Preprocessing

The study made use of two dataset. The first dataset was derived from Openphish while the other dataset was derived UCI. The Openphish dataset was generated in July 2025 and was requested and released for this study in August 2025. It comprised of 18 features. The dataset is mainly financial dataset from urls of organizations involved in offering financial services to the public; they are either into

cryptocurrency or payment platforms that entails users entering their financial details or credentials to carry out financial transactions of various forms.

UCI stands for the University of California, Irvine (UCI). The name of the UCI dataset for this study is the "Phishing Websites Features Classification" dataset, by [27] The UCI Repository Name is Phishing WebsitesKey.; this dataset has 30 features and 11,055 rows/instances. As stated, the UCI dataset has exactly 30 features plus one class attribute (Total of 31 columns).

The OpenPhish dataset used in this study was specially requested from OpenPhish in August, 2025 though the dataset was mined in July 2025. The UCI Phishing Websites dataset with ID 327 on the UCI Machine Learning Repository was generated by [27] but it was not released to general populace until 2015.

The OpenPhish dataset was in categorical form; the dataset was converted to numeric form as the model would only accept numeric value while the UCI dataset was already in numeric form. The IP and URL features of the Openphish dataset were one-hot encoded while other features were label-encoded.

The mathematics: Threshold-Based Phishing Detection

Ensemble phishing score:

$$S(x) = \alpha S_{AE}(x) + \beta S_{GAN}(x) + \gamma S_{SOM}(x) \quad (3.1)$$

Threshold t (95th percentile on benign training data for this study)

Decision rule:

$$x \text{ is } \begin{cases} \text{Phishing} & \text{if } S(x) > \tau \\ \text{Benign} & \text{if } S(x) \leq \tau \end{cases}$$

Table 3.1: Individual Model Anomaly Scores

Model	Score Formula	Notes
Autoencoder (AE)	$S_{AE}(x) = \ x - \hat{x}\ _2$	Reconstruction error
GAN	$S_{GAN}(x) = 1 - D(x)$	Discriminator likelihood
SOM	$S_{SOM}(x) = \min_k \ x - w_k\ _2$	Quantization error

Normalization was done to [0,1] before ensemble fusion
AE + GAN + SOM Ensemble Fusion Formula

$$S(x) = \alpha \tilde{S}_{AE}(x) + \beta \tilde{S}_{GAN}(x) + \gamma \tilde{S}_{SOM}(x) \quad (3.2)$$

and

$$\alpha + \beta + \gamma = 1 \quad (3.3)$$

Equation 3.6 combines reconstruction, distribution and topology signals.

F. Data normalization

The min-max normalization method is applied research applied in this research to convert the numerical column values in the OpenPhish and UCI datasets to a standard scale between 0 and 1 after converting from categorical data to numeric values and performing smart feature engineering without distorting the value ranges. The Equation applied is given as:

$$Y = a - \min(a) / \max(a) - \min(a) \quad (3.4)$$

Y = normalized value and
a = original value.

The intent of normalization is to ensure that all scores are comparable since the range and scale differ for each model ensemble and cannot be combined directly in the ensemble. The scaling is otherwise defined as in equation 3.5.

$$\tilde{s}_i = \frac{s_i - s_{\min}}{s_{\max} - s_{\min}} \quad (3.5)$$

Where:

s_i = raw anomaly score (AE, GAN, or SOM) for sample i

$s_{\min}, s_{\max}$ = min and max scores across training (benign) data

$\tilde{s}_i \in [0, 1]$ = normalized score

All normalized scores are in [0, 1], ready for weighted ensemble fusion:

$S(x) = \alpha \tilde{s}_{AE} + \beta \tilde{s}_{GAN} + \gamma \tilde{s}_{SOM}$, previously stated in equation 3.1 above.

AE + GAN + SOM Ensemble Algorithm

Step 1: Autoencoder Training

Train the Autoencoder on X by minimizing reconstruction loss: $\mathcal{L}_{AE} = \|x - \hat{x}\|_2^2$

Step 2: GAN Training

Train the Generator and Discriminator adversarially on X :

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log (1 - D(G(z)))]$$

Step 3: SOM Training

Train the Self-Organizing Map using competitive learning on X , updating neuron weights via neighborhood adaptation.

Step 4: Score Normalization

Compute training anomaly scores:

- AE reconstruction errors s_{AE}
- GAN discriminator scores s_{GAN}
- SOM quantization errors s_{SOM}

Normalize each score to $[0, 1]$:

$$\tilde{s}_m = \frac{s_m - \min(s_m)}{\max(s_m) - \min(s_m)} \quad \forall m \in \{AE, GAN, SOM\}$$

Step 5: Threshold Calibration

Determine anomaly threshold τ using a percentile of ensemble scores on training data.

Step 6: For each unseen sample x_j do

a. Compute AE reconstruction score:
 $s_{AE}(x_j) = \|x_j - \hat{x}_j\|_2$

b. Compute GAN anomaly score:
 $s_{GAN}(x_j) = 1 - D(x_j)$

c. Compute SOM quantization error:
 $s_{SOM}(x_j) = \min_k \|x_j - w_k\|_2$

d. Normalize scores: $\tilde{s}_{AE}, \tilde{s}_{GAN}, \tilde{s}_{SOM}$

e. Compute ensemble anomaly score:
 $S(x_j) = \alpha \tilde{s}_{AE} + \beta \tilde{s}_{GAN} + \gamma \tilde{s}_{SOM}$

Step 7: If $S(x_j) > \tau$ then

Denote x_j as Phishing
Else

Denote x_j as Benign

8. End

G. Model Validation: train/test split

Validating the model involves splitting the dataset into two sets namely training and testing datasets in the order 80/20; 80% for the training of the model while 20% of the dataset is for testing or model evaluating for performance measurement [29].

H. Model performance evaluation metrics

As applied by [9], the following quality measures were also used in this study and they are Accuracy, Precision, Recall and F1-Score in order to evaluate the performance of the unsupervised ensemble machine learning model developed in this study. Also, as explained by [27], phishing detection and prevention models must be evaluated and compared to a predetermined framework using the performance indicators such as accuracy and F1-Score: These measures will provide a thorough understanding of each model's performance in detecting phishing assaults [9].

$$TPR(\tau) = \frac{\# \text{ of phishing URLs with } S(x) > \tau}{\text{Total phishing URLs}} \quad (3.6)$$

$$FPR(\tau) = \frac{\# \text{ of benign URLs with } S(x) > \tau}{\text{Total benign URLs}} \quad (3.7)$$

where FPR = fraction of benign URLs above t
where t = threshold

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.8)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.9)$$

Recall (True Positive Rate)

Fraction of existent phishing URLs found:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.10)$$

F1-Score

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.11)$$

I. The Developed Model's Architecture

The model of this research is based on phishing threat alone so as to evaluate its performance compared with the existing models referenced in the literature review. This will fill the research gap as explained in literature review section.

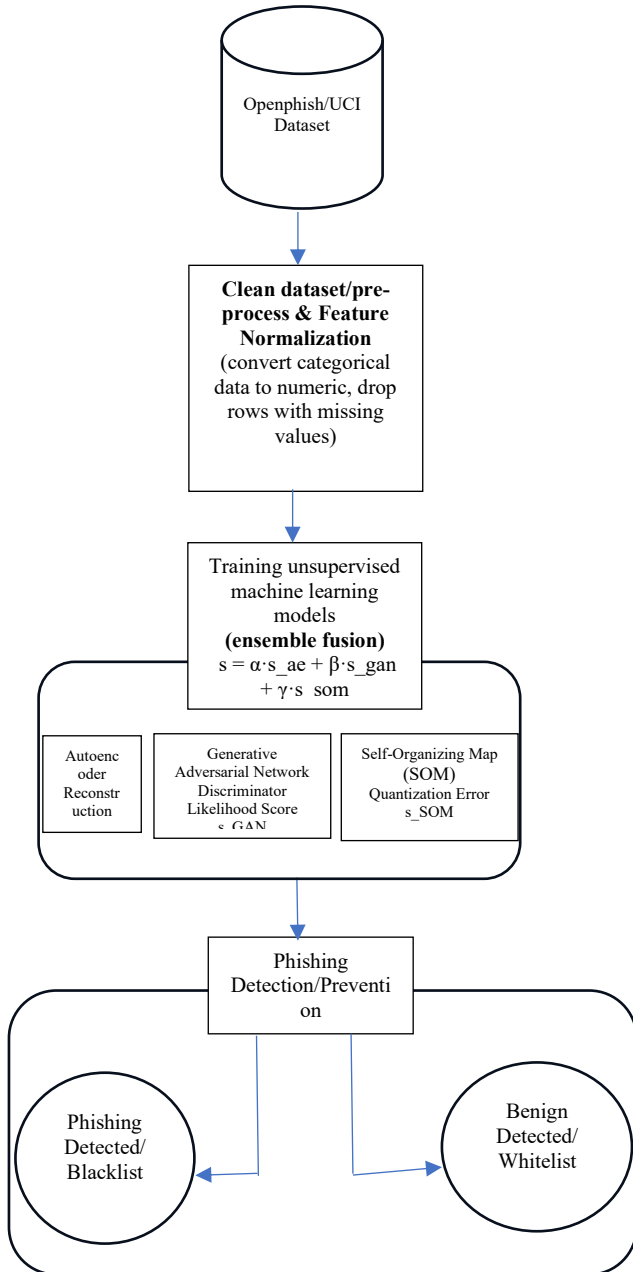


Fig. 3. The Developed Model's Architecture

Fig. 3 is the model's architecture built in this study. The figure explains that datasets were generated from twain sources, which are OpenPhish and UCI. These datasets have been described previously. The next stage shown in the figure is the preprocessing phase of the data, in which the OpenPhish categorical dataset was converted to numeric using one-hot encoding and label encoding; the OpenPhish dataset contained 18 features with 2141 rows/samples, while the UCI dataset was numeric and contained 30 features and 11, 055 rows/samples.

IV. RESULTS AND DISCUSSION

The experiment setup for the model built and the various results derived from the two datasets used in this study are discussed here. The study used twain datasets from OpenPhish, which was generated in July 2025 but requested by the author of this thesis in August 2025 and released in the same month, as well as the UCI dataset, which was generated in 2015 and was released for scholarly use in 2015 but was collected by the author of this work from the machine learning repository of the University of California, Irvine (UCI).

A. Experiment Setup and System Development

This aspect fulfils the second objective of the study. Here, the model designed in the system architectural diagrams and algorithms are implemented. It deals with the actual coding that resulted in the model developed.

The OpenPhish and UCI datasets were compared; the two datasets were collected for the experiment in order to verify the variability of results from two or more dataset sources.

Fig. 4 shows the setup for the whole experiment. The experiment starts with training the datasets from the two sources, OpenPhish and UCI. Once the data is trained, the artifacts are then fed into FastAPI code which detects which URL is benign or phishing. From here, a man-in-the-middle certificate authority software from Microsoft is installed, and proxy settings are configured on port 8080; the dataset in CSV format can be tested on the FastAPI, and predictions are saved in a CSV format, which can then be downloaded. MITM script will control the operation of the proxy server.

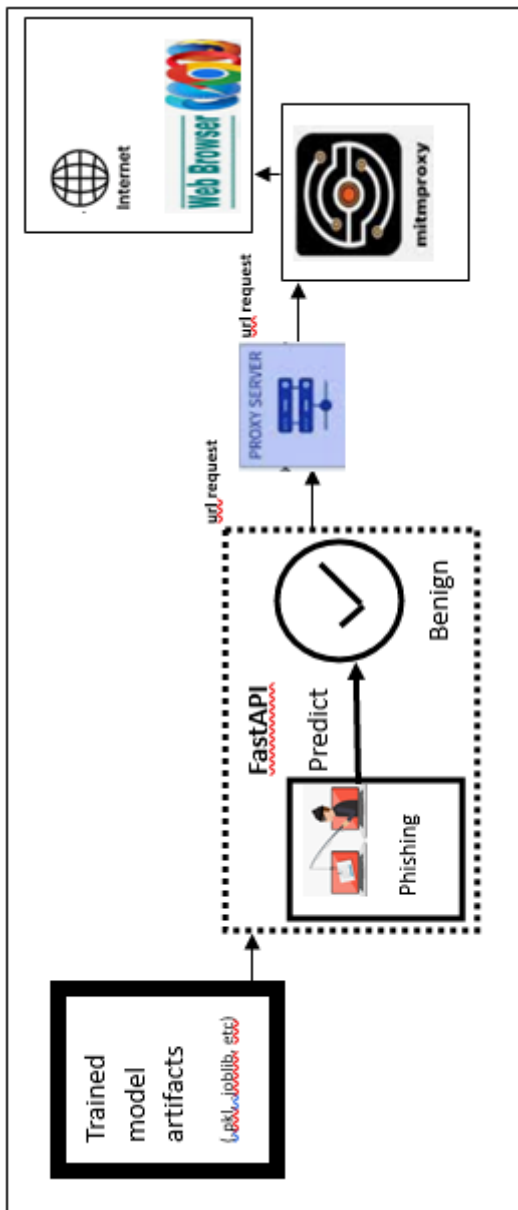


Fig. 4. Setup of the research experiment for detecting and preventing phishing URLs

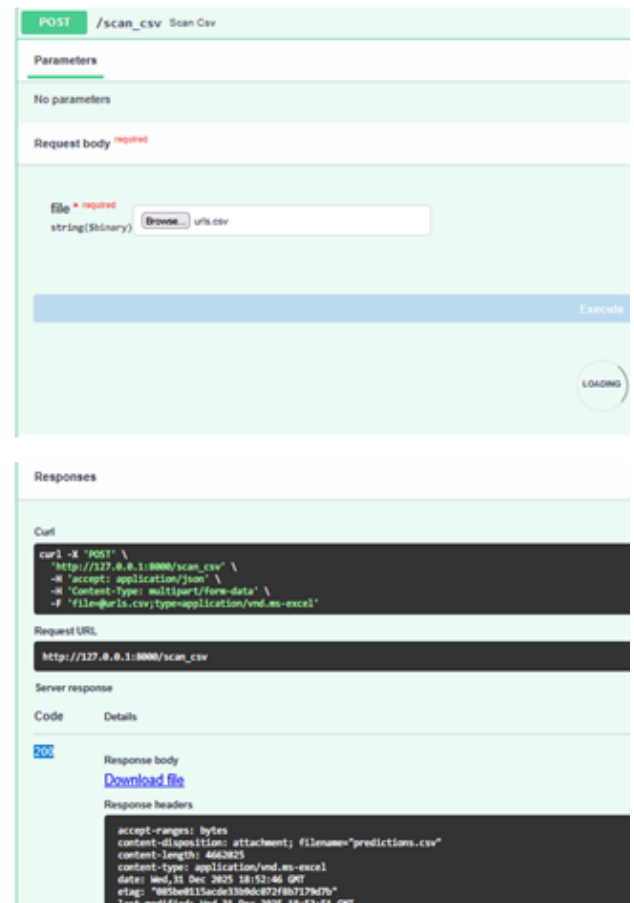


Fig. 5. Displays the response of the bashing of the FastAPI script in the browser. The FastAPI/Swagger is accessed using “127.0.0.1:8000/docs”.

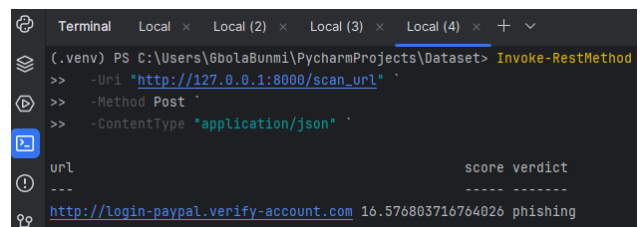


Fig. 6. Response from a single URL curl in PowerShell/ Pycharm terminal.

Fig. 6 shows the response of PowerShell Invoke for single URL testing.

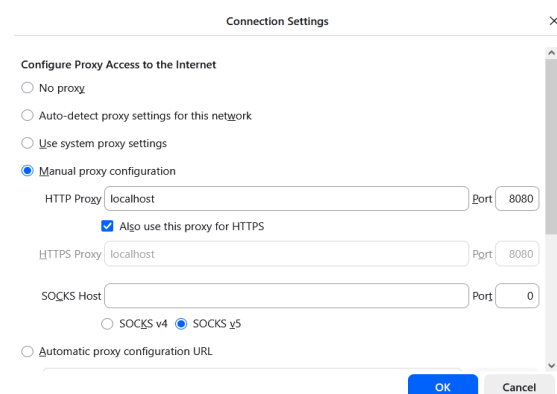


Fig. 7. Enabling Proxy Server

Fig. 7 shows the proxy setting on the system used for carrying out the experiment for the research. This configuration must be done once mitm certificate authority is installed, as previously explained in this chapter. This proxy setting is done before running the MITM script so as to connect with it once the mitm listens to port 8080 as put up while installing the MITM Certificate Authority software.

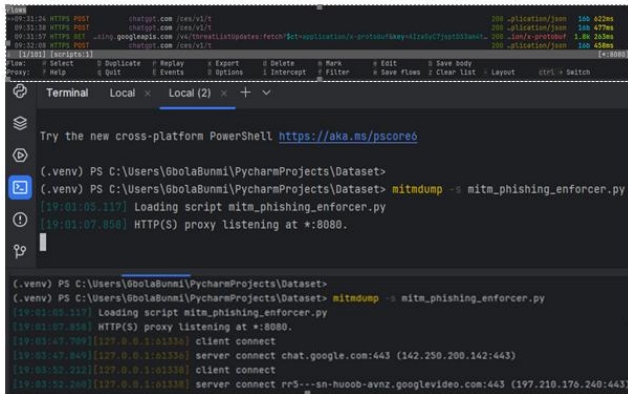


Fig. 8. Displays the response to MITM script bashing.

This script, in conjunction with the FastAPI server, prevents phishing URLs from opening and allows benign URLs to open. This marks the fulfillment of the third objective of this study. That is, an unsupervised ensemble model of AE + GAN and SOM has been built which efficiently prevents phishing of financial data in real time. The following script was used for bashing the MITM script written for this model in the PyCharm terminal: “mitmdump -s mitm_phishing_enforcer.py”; this actually depends on what you name your file.

The following URL samples were tested in the browser in order to authenticate the model and to know how efficient the model is; this marked the completion of the model.

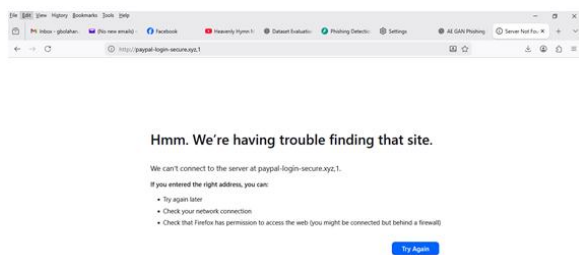


Fig. 9. shows the response of the bashing of the MITM script as previously explained in Fig. 8.

This Fig. 9. captures the result returned when a phishing URL is blocked from opening. A benign URL opens normally without interruption. Strong protection is provided by the browser block, which often stops the HTTP request from fully processing.

B. Results (Analysis)

The various outcomes from the experiment carried out are discussed here. The charts used and their interpretations are done in this section.

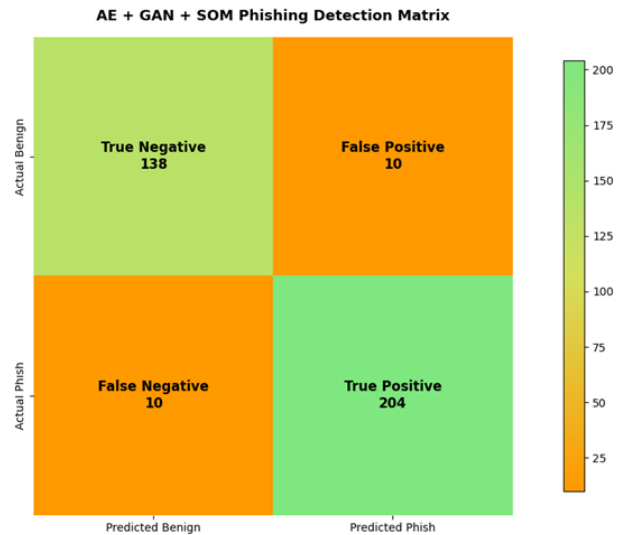


Fig. 10. illustrates the confusion matrix for the OpenPhish model.

362 observations were used due to missing values and the imbalanced nature of the dataset. Out of 362 instances eventually used, 138 were true negative which are number of benign URLs correctly identified (True Negative); 10 instances were false positive that were number of legitimate URLs inaccurately flagged as phishing (False Positive); only 10 instances are false negative, it means the number of phishing URLs missed (False Negative) and finally, 204 instances were number of phishing URLs correctly detected (True Positive).

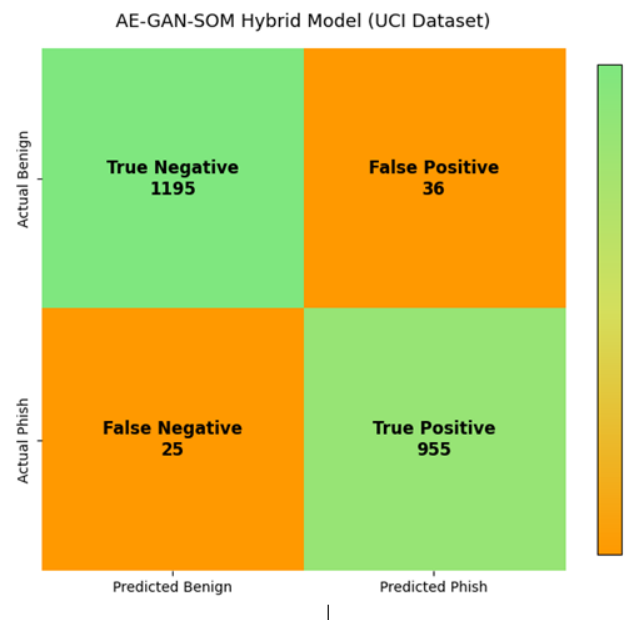


Fig. 11. Confusion Matrix from the Model using UCI dataset

Fig. 11 is an illustration of confusion matrix for model from UCI dataset. Out of 2210 instances, 1195 were true negative which are number of benign URLs correctly identified (True Negative); 36 instances were false positive which total benign URLs inaccurately flagged as phishing (False Positive); only 25 instances were false negative which means the total phishing URLs missed (False Negative) and

finally, 955 instances were total phishing URLs accurately detected (True Positive).

This confusion matrix shows operational behavior of the model at deployment threshold. The performance of the model developed is gauged using a confusion matrix. It is an illustration of point in which the model was accurate or inaccurate which is carried out simply by contrasting its predictions with the actual outcomes. It assists in visualizing how the model performs and potential error zones.

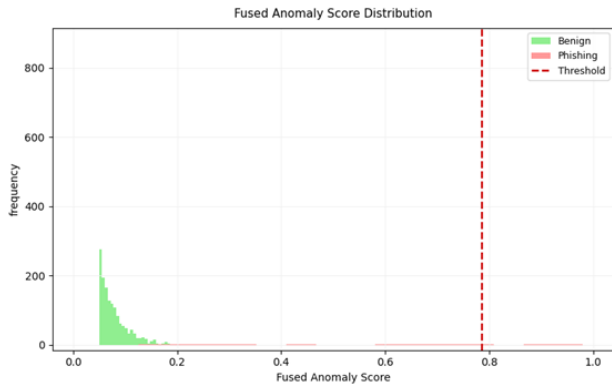


Fig. 12.

Fig. 12 shows the graph of models from the OpenPhish dataset anomaly score distribution. Anomaly score learning is a technique that uses statistical rarity and semantic variances to assign numerical scores to data occurrences. It combines deep representation, density-based, and margin-based methods to detect abnormalities in a variety of domains with robustness. To increase detection accuracy and severity assessment, the method makes use of multimodal fusion, regularization, and principled objective functions.

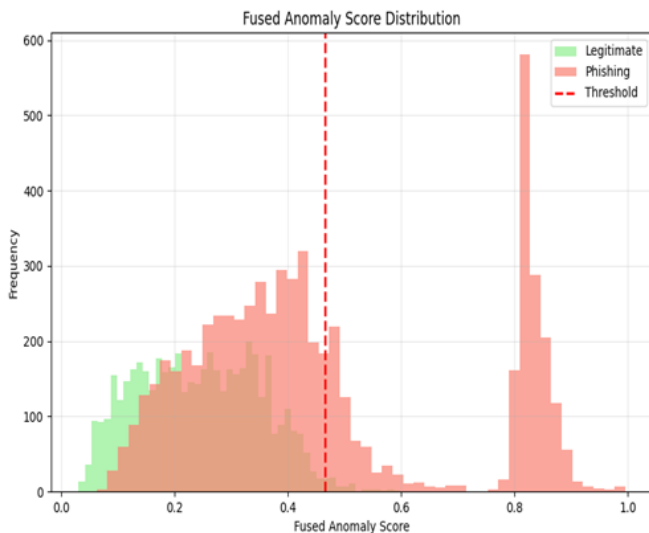


Fig. 13. UCI dataset Anomaly Score Distribution

Fig. 13 shows the graph of anomaly score distribution for the model from the UCI dataset. This fused anomaly score distribution is actually the most important plot among the visualizations and shows a histogram depicting the resultant ensemble anomaly score:

$$S(x) = \alpha \tilde{s}_{AE} + \beta \tilde{s}_{GAN} + \gamma \tilde{s}_{SOM} \quad (3.7)$$

with X-axis depicting the normalized anomaly score (0 to 1) while the Y-axis depicts the density/frequency. Those green bars/curves indicate legitimate/benign urls while the pink bars/curves indicate phishing urls. However, the pink vertical line shows that decision threshold t as used in the predictions. With this, the benign urls are separated from the phishing urls. Further interpretation of the chart shows benign urls scores clustered close to zero while phishing scores drifted towards one. With the low scores, we have normal web behavior while with the high scores, we have suspicious lexical patterns. One last thing worth mentioning is the overlapping region between the phishing and benign curves/bars; this is called the gray-zone urls.

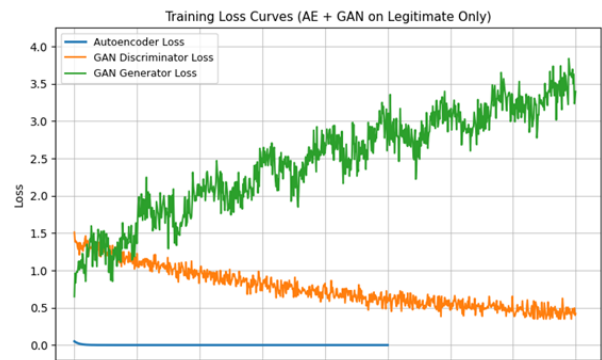


Fig. 14. shows the training loss curve for the model from the Openphish dataset.

From the visuals, it is established that the Openphish dataset was less balanced than the UCI dataset. Openphish contained more noise than the UCI dataset, which accounted for the significant loss in the Openphish training data. The training loss curve is for the autoencoder and generative adversarial network; UMAP distance is for the self-organizing map. For the GAN, there is a balance between D and G losses for the datasets used. In summary, the training loss curves indicate stable convergence of the autoencoder model and balanced adversarial learning in the GAN, thereby ensuring the anomaly score is reliably estimated.

Plotting the error rate (y-axis) of the GAN + AE + SOM ensemble hybrid model against training iterations or epochs (x-axis- an epoch is a single execution of the whole training dataset; at this point, each data sample is run through the model and its parameters are modulated in reaction to the estimated error) is known as the training loss curve. It shows how well a model learns; improvement is indicated by a downward-sloping curve. It aids in the diagnosis of problems such as underfitting or overfitting (low training loss, high validation loss). The difference between the expected and actual output is measured by the loss function. When the loss curve is decreasing, the model is effectively lowering mistakes with time. It assists in tracking the learning process, enabling practitioners to assess if the model is overshooting because of a fast-learning rate, convergent too slowly, or learning well.

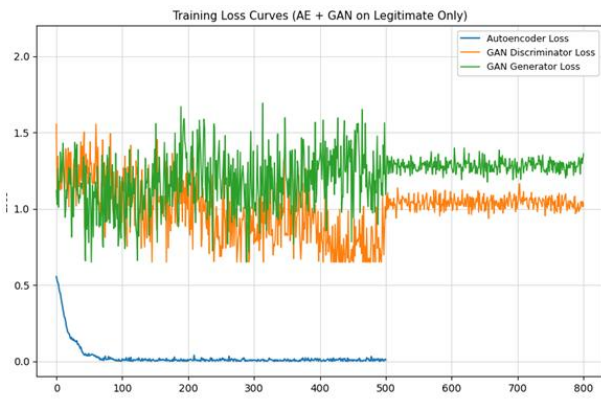


Fig. 15. Training loss curve of UCI dataset

Fig. 15 shows the training loss curve for both the OpenPhish dataset and UCI dataset. From the visuals, it is established that the OpenPhish dataset was less balanced than the UCI dataset. OpenPhish contained more noise than the UCI dataset, which accounted for the significant loss in the OpenPhish training data. The training loss curve is for the autoencoder and generative adversarial network, u-map distance is for the self-organizing map. For GAN, there is a balance between D and G losses for the datasets used. In summary, the training loss curves indicate stable convergence of the autoencoder model and balanced adversarial learning in the GAN, thereby ensuring the anomaly score is reliably estimated.

When the graph is descending, it means learning is happening in the model. However, when it is flat, it is either because the learning speed is too low or the built model has converged (stopped learning). Lastly, when it is fluctuating, the batch size may be too small or the learning rate excessively high. High loss indicates that the data may not have been adequately preprocessed or the model may not have enough capacity.

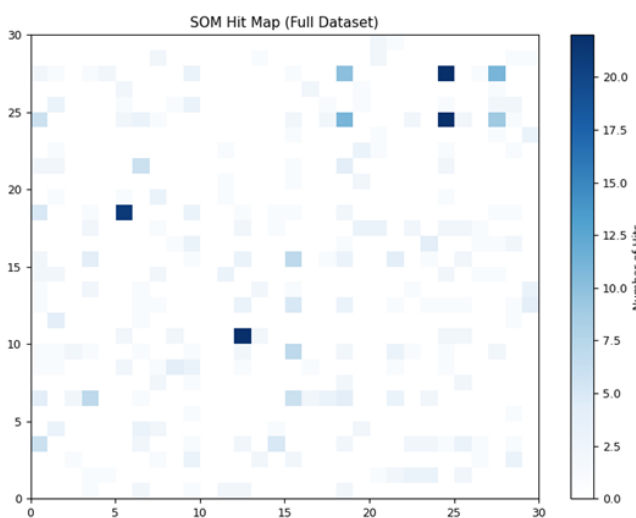


Fig. 16. SOM hit map for Openphish Dataset

Fig. 16 shows the SOM hit map for Openphish datasets. The SOM Hit Map is also known as the Activation Frequency Map. The chart shows a grid of SOM neurons, with each cell showing how often it was activated. The bright blue colors

show frequently activated neurons while the dull blue colors show rarely activated neurons. This chart is a density estimation, and the bright blue colors, or high-hit regions, show where benign URLs dominate and mean normal web behaviour, while the dull blue colors, or sparse regions, show phishing URL domination. In this case, phishing is rare and scattered.

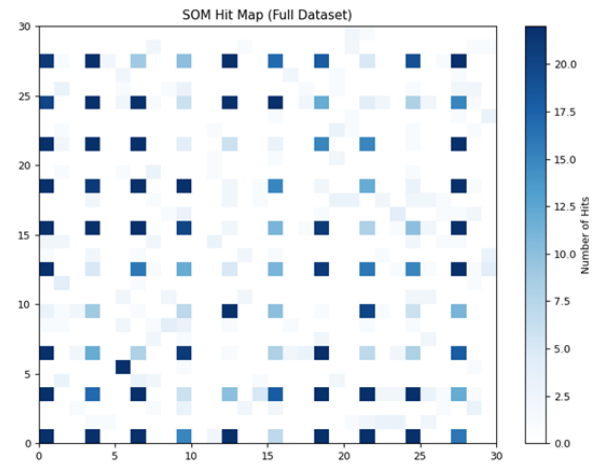


Fig. 17. SOM hit map for UCI dataset

Fig. 17 shows the SOM hit map for the model from the UCI datasets. The SOM Hit Map is also known as the Activation Frequency Map. The chart shows a grid of SOM neurons, with each cell showing how often it was activated. The bright blue colors show frequently activated neurons, while the dull blue colors show rarely activated neurons. This chart is a density estimation, and the bright blue colors, or high-hit regions, show where benign URLs dominate, meaning normal web behaviour, while the dull blue colors or sparse regions, show phishing URL domination. In this case, phishing is rare and scattered.

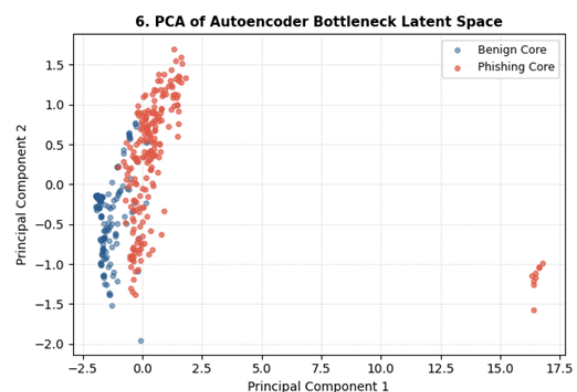


Fig. 18. PCA Visualization for model using Openphish Dataset

Fig. 18 shows the PCA visualization of the model from Openphish. These charts display the PCA visualization for each model's latent space. But this chart displays the PCA visualization of the autoencoder. Typically, PCA projects high-dimensional latent vectors into 2 dimensions. Each point plotted is equivalent to one URL. PCA prediction of the learned latent spaces shows that phishing URLs consistently

occupy outer or low-density regions, thus validating the representational capacity of the model.

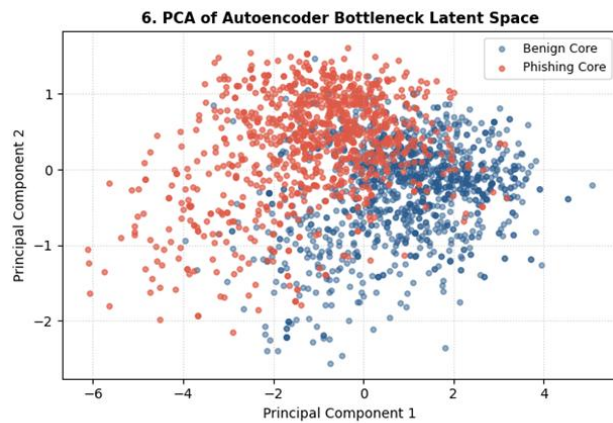


Fig. 19. PCA Visualization for model using UCI Dataset

Fig. 19 shows the PCA visualization of the model from UCI datasets. These charts display the PCA visualization for each model's latent space. But this chart displays the PCA visualization of the autoencoder. Typically, PCA projects high-dimensional latent vectors into 2 dimensions. Each point plotted is equivalent to one URL. This is just a representation validation. In the autoencoder PCA, the AE compresses the normal structure. It shows that phishing violates learned reconstruction patterns. PCA prediction of the learned latent spaces shows that phishing URLs consistently occupy outer or low-density regions, thus validating the representational capacity of the model."

C. Performance Evaluation of the models from OpenPhish and UCI datasets

The results from the testing done with 20% of the dataset are discussed here. The confusion matrix, as illustrated in Figures 9 and 10, displays the four quadrants, explaining the true phishing and the number of benign URLs. The remaining part of this section compares models from the two datasets as well as compares the models from the two datasets with all other models reviewed in the literature.

Table 1. Evaluation metrics of Openphish and UCI datasets

Metrics	Openphish dataset (%)	UCI dataset (%)
Precision	95.33	96.37
Recall	95.33	97.45
F1 Score	95.33	96.91
Accuracy	94.48	97.24

Table 1 shows that Openphish Dataset with accuracy of 94.48% performed below model from UCI dataset with accuracy of 97.24%. The features from both datasets are relevant for modern phishing/anomaly detection and prevention. This is further revealed in the fused anomaly score distribution of the model from the UCI dataset when compared to that of the model from the OpenPhish dataset.

D. Comparison of the Developed Model with Other Models in the Literature Review

The outcome of the model built from two different datasets namely Openphish and UCI indicated that the model from Openphish dataset has accuracy of 94.48%, 95.33% precision and 95.33% recall compared to the model from UCI dataset with accuracy of 94.24%, 96.37% precision and 97.45 recall of model from UCI dataset Overall, the model from UCI dataset widely outperformed the model from Openphish dataset; this was further revealed in the visualizations of models from both datasets as previously shown and explained in this study. Models from each dataset exceeded the acceptable detection rate.

Table 2. Comparison of the Results with Existing Models

S/N	Method	Accuracy	Author
1.	Deep Multi-Class Neural Network vs our Hybrid Unsupervised Ensemble	95.32% as our higher accuracy of 97.24%	Reddy <i>et al.</i> (2022)
2.	Supervised Deep Learning Standard (NN), (Conv2D) vs our Hybrid Unsupervised Ensemble	94.80% (Standard NN), 93.60% (Conv2D) as against our accuracy	Siddiq <i>et al.</i> (2022)
3.	Logistic Regression, Support Vector Machine and Random Forest.	Accuracy of 83.83%, 82.30% and 75.80% respectively as against ours of higher accuracy of 97.24%.	Broštic <i>et al.</i> (2025).
4.	Supervised DL/ML Models (1D-CNN), (K-NN), (LSTM) vs our Hybrid Unsupervised Ensemble	96.75% (1D-CNN), 94.34% (K-NN), 91.23% (LSTM) vs our higher accuracy of 97.24%	Al Mahmud <i>et al.</i> (2022)
5.	ANNs, support vector machines (SVMs), decision trees (DTs), and random forests (RFs)	DT at 96.3%, ANN at 95.5%, SVM at 94.66% lower compared to our accuracy of 97.24%	Alnemari and Alshammari (2023)
S/N	Method	Accuracy	Author
6	Support Vector Machines,	Highest overall classificatio	Alzboon <i>et al.</i> (2025)

	Decision Trees, and Random Forests vs our Hybrid Unsupervised Ensemble (AE + GAN + SOM)	n capacity at 97% accuracy, almost at par with our accuracy, yet a bit lower in comparison with out accuracy of 97.24%		(AE + GAN + SOM)	
7	Supervised Heuristic-driven traditional ML vs our Unsupervised Ensemble (AE + GAN + SOM)	Accuracy of 97.2% (URL) / 98.1% (Website) vs our 97.24% (Stable across financial streams)	Jayaprakash <i>et al.</i> (2024)	26 Support Vector Machines (SVM), Random Forest, Multilayer Perceptron (MLP) and XGBoost vs our Unsupervised Ensemble (AE + GAN + SOM)	XGBoos t accuracy is 91% , MLP and Random Forest classifiers are 91%, Decision Tree at 90%. Logistic Regression and SVM accuracies are 87% and 88%; all their accuracies are lower than our accuracy of 97.24%
8	Three specific techniques: Logistic Regression, Multinomial Naive Bayes, and XGBoost vs our Unsupervised Ensemble (AE + GAN + SOM)	Peak accuracy of 96.63% lower than our 97.24% accuracy.	Neeraja <i>et al.</i> (2024)		
9	Random Forest ensemble and Decision Tree classifier vs our Unsupervised Ensemble	Accuracy rate of 87.0% Random Forest ensemble and a lower accuracy of 82.4% Decision Tree classifier	Mandadi <i>et al.</i> (2022)		
S/N	Method	Accuracy	Author		
25	Support Vector Machines (SVM) and Random Forests (RF) and Convolutional Neural Network (CNN) vs our Unsupervised Ensemble	64%, 67% and 70% accuracy respectively	Varghees Sreelakshmi (2022)		

Table 2 shows the comparison of the model developed in this study with the models from the literature review; it was discovered that the accuracy of our model is far over 90% of the models reviewed in the literature. This is a great achievement for our study; this makes our model more valid than the other models reviewed in the literature review.

E. Discussion Of Findings

This study is a huge success; what seemed like an uphill task with Herculean heels of sourcing for recent financial datasets was overcome with a well-developed model for phishing detection and prevention.

The first objective to design a hybrid unsupervised ensemble architecture integrating deep reconstruction, adversarial density, and topological clustering, was accomplished in the system architectural structures, which included the theoretical framework, conceptual framework and the developed model architecture.

The second objective was effectively achieved; the study developed a hybrid unsupervised ensemble model integrating Autoencoders (AE), Generative Adversarial Networks (GAN), and Self-Organizing Maps (SOM) for real-time detection. It was a great achievement, as the researcher was able to solve a problem. Phishing as a vulnerability is fully combated using the model developed in this study; the banking sector as well as other professions can apply this model, as it has proven to be successful through the metrics used in evaluating the model. The hybrid approach effectively addresses the drawbacks of single-algorithm model by fusing their individual strengths. It was discovered that a hybrid of three or more models is highly effective in detecting and preventing phishing from taking place.

The greatest achievement came when the model was tested with real-life urls in real-time and it was able to block phishing urls successfully while benign urls were allowed to open. The model was able to implement a real-time traffic

interception and processing pipeline using FastAPI and MITM proxy configured with a Firefox browser proxy to facilitate live analysis of URL traffic and restraining phishing attacks in financial sector. Thus, the third objective was accomplished.

The last objective was achieved when the model developed from both datasets were validated; using evaluation metrics earlier stated, the models from the two datasets were compared and the result showed that the model from the Openphish dataset performed better than the model from the UCI dataset. The research was able to further evaluate the model developed with feat of the models from authors reviewed in the literature. It was found that the models from the study compare highly with the models from other studies. Our model supersedes models from other studies reviewed in the literature with a whopping 97.24%.

It was found that ensemble unsupervised is better for preventing phishing than single supervised or single unsupervised ensemble model. Likewise, it was discovered from the comparison of the model from this study and the models from the reviewed literatures that combination of deep learning techniques is better off using combination of artificial neural network and deep learning.

What the financial systems need is detection of a Zero-day attack; zero-day attack often takes vantage a software or hardware vulnerability which is barely known to the vendor or internet users, thereby leaving "zero days" for defenders to create a patch for bolstering the vulnerability. The model has achieved that. The financial system will also benefit low false positive rate so as to avoid blocking legitimate customers from using their websites. Not only that, the financial systems will also have a millisecond-level response as well as adaptability to new phishing patterns.

No study so far has combined autoencoder, generative adversarial network and self-organizing map, only this study has done that, this means that this study is a major contribution to the field of machine learning and cybersecurity. The synergy among AE, GAN and SOM is a positive for this study. The findings suggest that the score-level fusion of three distinct unsupervised signals significantly outperformed any single-model approach.

The study found that calibrating the anomaly threshold at the 95th percentile of the benign distribution provided an optimal balance between security and user experience. By defining "normal" so strictly, the model successfully isolated the most aggressive structural and behavioral deviations. The 95th percentile acts as a conservative "security-first" boundary necessary for high-stakes banking environments.

V. CONCLUSION AND SUGGESTION FOR FUTURE STUDIES

A. Summary

The study reviewed literatures on different techniques, both supervised and unsupervised that is used for detecting and blocking phishing. Some of the solutions proffered used single model while some other used hybrid techniques, with few using novel model when many used ensembles model, a combination of different models which is more efficient than the single models reviewed. The issue of phishing as cybercrime is evolving and taking new dimensions as

technology advances. The only way to secure the cyberspace is to fully involve experts in the field of information technology/computing; these professionals would write computer programs to counter these crimes that is making government, corporate entities and individuals to lose assets in trillions annually and making susceptible victims to commit suicide. This section accomplishes the first objective of the study.

In summary, the study adopted a hybrid unsupervised ensemble methodology which combined Autoencoders, Generative Adversarial Networks, and Self-Organizing Maps to model benign URL pattern, detect and block/prevent phishing attempts as statistical anomalies. The methodology was divided into design and development; the design aspect took care of the conceptual and theoretical framework (architecture/blueprint), while the development took care of the algorithms, mathematical models, the logic, the libraries applied in developing the proposed model. The outcome is a standard model that is able to identify and prevent real-time phishing in the financial sector.

B. Conclusion

This study reviewed relevant literatures on the real-time prevention of phishing in the financial sector. Financial data is crucial to both the users and the players in the financial sector. These data are often sought after by unscrupulous people called the phishers who lurk around on the webspace, the dark world trying to find victims who they can easily defraud by gaining access to their financial credentials via a cloned website that is similar to the original from the financial company. Users of internet often fall victims because the website is a replica of the cloned site and can only be differentiated if the user is patient enough or already trained on possibility of such occurring.

The study applied ensemble behavioural anomaly detection for adversarial url threats which can be applied in preventing malware urls and fraud links. Finally, just as this model works in real-time using FastAPI, the model can also work offline to scan logs of phishing urls saved in csv format, most especially.

The developed model is very secured for deployment. The threat of phishing should not be taken for granted. The study deployed unsupervised ensemble model to detect and prevent phishing; the combination of GAN, AE and SOM outperformed most previous models that attempted to detect phishing. Using MITM certification authority aided the prevention of phishing urls. All phishing urls were blocked.

The initiative to use deep learning with artificial neural network which made the ensemble model highly efficient in detecting and preventing phishing urls in real-time with accuracy of 97.24%. It was noted that ensemble model of deep learning and artificial neural network is the future of anomaly detection that researchers can even use in novel approach, hence it is suggested that future studies should apply the ensemble model used in this study for deepfake detection and prevention. There is great hope in the hybrid model developed in this study, it believed that if larger features and instances are used for the combination of AE + GAN + SOM, there will be better results than what was achieved here. If the banking sector can adopt the model

developed, the key players in this sector can relax and watch as technology is used to perform what human beings cannot ordinarily perform manually.

C. Recommendations and Suggestion for future Studies

Based on the successful outcome of this study, it is suggested that further researches use the ensemble method used in this research work to detect and prevent e-mail spamming. Future study should consider using deep and artificial neural network to develop software that can be installed permanently on computers and mobile devices.

The first aid to the cyberspace menace is user education. All internet users should be well informed on the techniques used by phishers and how to recognize phishing urls so as to avoid them.

The financial sector also needs to educate employees on how to avoid phishing traps and the industry should employ experts in the field of machine learning and cybersecurity to oversee their system security. The financial industry should endeavor to spend on their system security; this will not only safeguard the banks; it will also safeguard the users of various banking products.

The study did well to use deep learning with artificial neural network which made the ensemble model highly efficient in detecting and preventing phishing urls in real-time with accuracy of 97.24%. It was noted that ensemble model of deep learning and artificial neural network is the future of anomaly detection that researchers can even use in novel approach, hence it is suggested that future studies should apply the ensemble model used in this study for deepfake detection and prevention.

Based on the successful outcome of this study, it is suggested that further researches use the ensemble method used in this research work to detect and prevent e-mail spamming.

Further study should also consider using ANN and deep learning to locate the exact location/spot the phisher is operating from; further features can be obtained from the URL and IP of the dataset used in such model.

D. Contribution to knowledge

This research contributes a real-time phishing prevention framework which ensembles heterogeneous unsupervised anomaly detection models without relying on labeled phishing feature in the dataset. Furthermore, the model fills the gap of this research by building a model that ensembles both deep and artificial neural networks models to prevent phishing of financial data in real-time. This system is deployable in real-time. Most phishing detection models are limited by very false positives on legitimate urls but this study addressed that issue.

Using dataset from Openphish and UCI reduced dataset biases. It shows that a dataset can have a very high accuracy and still perform below a dataset with lower accuracy but higher precision and recall. Above all, ensemble of multiple ANN and Deep Learning models yields more robust detection than single model. It should be noted that the researcher also tested individual models separately and came to the conclusion that ensemble model of ANN and Deep Learning outperform single model. This ensemble model

detects phishing urls not seen during training. This study also contributes to research methods in machine learning and cybersecurity.

Acknowledgment

I appreciate Openphish for the dataset released for the purpose of this study. The model training script, FastAPI script, MITM Proxy Script and the testing script for model from Openphish dataset will be made available alongside the training script and the testing script for model from UCI dataset once the manuscript is accepted for publication, if required.

REFERENCES

- [1] S. Pandey and M. Kumar. (2023). Cybersecurity trends and challenges. *International Journal of Scientific Research in Engineering and Management (IJSREM)* Volume: 07 Issue: 08, August.
- [2] S.N. Romaniuk and M. Manjikian (Eds.). (2021). Routledge companion to global cyber-security strategy (1st ed.). *Routledge*.
<https://doi.org/10.4324/9780429399718>
- [3] B. Vinisha, and S. Balasinorwala (2023). Cyber security. *International Journal of Scientific Research in Engineering and Management*, 7(2).
<https://doi.org/10.55041/IJSREM17691>
- [4] B.C. Tenuche, E. Agbata, E. Ogala and W. Obeng-Denteh (2020). The fuzzy experiment approach for detection and prevention of phishing attacks in online domain. *East African Scholars J Eng Comput Sci. Volume-3 Issue-10*. doi: 10.36349/easjecs.2020. v 03i10.001
- [5] P. Sornsuwit Ployphan and J. Saichon (2019). A New Hybrid machine learning for cybersecurity threat detection based on adaptive boosting, *Applied Artificial Intelligence*, 33:5, 462-482, doi: 10.1080/08839514.2019.1582861
- [6] A. Ajayi, M. Olajide, O. Abiodun and O. Afolabi (2024). Evaluation of phishing attack strategies on mobile device users. *International Journal of Scientific and Management Research*, 7(9), 137–148.
- [7] I.H. Sarker, S. Jamal and H. Wimmer (2024). An improved transformer-based model for detecting phishing, spam and ham emails: *A large language model approach*. *Security and Privacy*, 7(5), e402.
- [8] H. Zamil and M.O. Faruq (2022). Cybersecurity and data integrity in financial systems: A review of risk mitigation and compliance models. *International Journal of Scientific Interdisciplinary Research*. 1(1), 27–61.
<https://doi.org/10.63125/azwzvnv07>
- [9] A.P. Abidoye and B. Kabao (2022): Hybrid machine learning: A tool to detect phishing in communication network. *International Journal of Advanced Computer Science and Application*, vol. 11 No 6 pp. 559-569.
- [10] S. Chen, & W. Guo (2023). Auto-encoders in deep learning: a review with new perspectives. *Special Issue Mathematical Methods and Applications for Artificial Intelligence and Computer Vision*, 11(8), 1777. Doi: 10.3390/math11081777
- [11] T. Chakraborty, U. R. K. S Reddy, S. M. Naik, M. Panja

- and Manvitha, B. (2023). Ten Years of Generative Adversarial Nets (GANs): *A survey of the state-of-the-art*. arXiv:2308.16316v1 [cs.LG] 30 Aug
- [12] L. Khacef., L. Rodriguez. and B. Miramond (2020). Improving self-organizing maps with unsupervised feature extraction. *Universite C'ote d'Azur, CNRS, LEAT, France*
- [13] E. S. Shombot, G. Dusserre, R. Bestak, and N. B. Ahmed (2024). An application for predicting phishing attacks: A case of implementing a support vector machine learning model. *Cyber Security and Applications* 2. 100036
- [14] D. Komalasari, T. B. Kurniawan, D. A. Dewi, M. Z. Zakaria, Z. Abdullah, and A. Alanda (2025). Phishing domain detection using machine learning algorithms. *International Journal on Advanced Science Engineering Information Technology*, Vol.15 No. 1
- [15] A. Y. Reddy, A. Gouni, and S. Akavaram (2022). AI based feature selection with unsupervised learning for efficient spam and phishing email classification. *International Journal of Communication Networks and Information Security*, 14(2)
- [16] M. A. A. Siddiq, M. Arifuzzaman, and M. S. Islam (2022). Phishing website detection using deep learning. In *2nd International Conference on Computing Advancements (ICCA 2022)*, March 10–12, 2022, Dhaka, Bangladesh. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3542954.3542967>
- [17] A. M. Broștic, B. A. Mărgărit, D. S. Mihaî, A. Nicolescu, and E. Simion (2025). A study on detecting and preventing phishing attacks using machine learning techniques. *Romanian Cyber Security Journal*. Fall, No. 2, Vol. 7
- [18] T. H. Al Mahmud, T. A. Khandaker, M. J. Uddin, M. M. Pias, and A. J. Rupok (2022). An automatic intelligent framework for a phishing detection system based on machine learning with urls features. *International Journal of Scientific & Engineering Research* Volume 13, Issue 8, August
- [19] S. Alnemari and M. Alshammari (2023). Detecting phishing domains using machine learning. *Appl. Sci.*, 13, 4649. <https://doi.org/10.3390/app13084649>.
- [20] M. S. Alzboon, M. S. Al-Batah, M. Alqaraleh, F. Alzboon, and L. Alzboon (2025). Guardians of the web: harnessing machine learning to combat phishing attacks. *Gamification and augmented reality*; 3:91. doi: 10.56294/gr202591
- [21] Jayaprakash R, Natarajan K, Daniel J.A, Chinnappan C.V, Giri J, Qin H and Mallik S. (2024) Heuristic machine learning approaches for identifying phishing threats across web and email platforms. *Front. Artif. Intell.* 7:1414122. doi: 10.3389/frai.2024.1414122
- [22] D. Neeraja, V. Ramana Murthy, D. R. S. Reddy, B. Rakesh, V. Swathika, and B. U. Kiran (2024). Detection of phishing websites using machine learning. *International Journal of Novel Research and Development*. Volume 9, Issue 3 March
- [23] A. Mandadi, S. Boppana, V. Ravella, and R. Kavitha (2022). Phishing Website Detection Using Machine Learning. *7th International Conference for Convergence of Technology (I2CT)*
- [24] D. M. Vargheese & N. R. Sreelakshmi (2022). Phishing website detection using machine learning techniques and CNN. *International Journal of Engineering Research & Technology (IJERT)*. Special Issue, Volume 10, Issue 04
- [25] M. W. Shaukat, R. Amin, M. M. A. Muslam, A. H. Alshehri, and J. Xie (2023). A hybrid approach for alluring ads phishing attack detection using machine learning. *Sensors*, 23, 8070. <https://doi.org/10.3390/s23198070>
- [26] B. Sarker, P. Zhang, A. Oest, H. Cho, Z. Sun, R. C. Johnson, S. Wardman, A. Kapravelos, T. Bao, R. Wang, Y. Shoshitaishvili, A. Doupe, and G. J. Ahn (2021). CrawlPhish: Large-scale analysis of client-side cloaking techniques in phishing. *IEEE Symposium on Security and Privacy (SP)*, 1109–1124. <https://doi.org/10.1109/SP40001.2021.00021>
- [27] R. M Mohammad, F. Thabtah and L. McCluskey (2012). An assessment of features related to phishing websites using an automated technique. In *2012 International Conference for Internet Technology and Secured Transactions (ICITST)* (pp. 492- 497). IEEE. doi: 10.24432/c51w2x
- [28] S. Kavya and D. Sumathi (2025). Staying ahead of phishers: a review of recent advances and emerging methodologies in phishing detection. *Springer: Artificial Intelligence Review*, 58:50 <https://doi.org/10.1007/s10462-024-11055-z>
- [29] K. Joshi, C. Bhatt, K. Shah, D. Parmar, J. M. Corchado, A. Bruno, & P. L. Mazzeo (2023). Machine-Learning Techniques for Predicting Phishing Attacks in Blockchain Networks: A Comparative Study. *Algorithms*, 16, 366. <https://doi.org/10.3390/a16080366>