HYBRID TRANSFORMER-BASED CLASSIFICATION FOR WEB-BASED INJECTION ATTACK DETECTION: A NOVEL ROBERTA-XLNET APPROACH

Ranuja Seethawaka, Chathurya Nambuwasam, D.K.W.G.G.T. Chandrasiri, K.A.S. Kavithma, Harinda Fernando and Ayesha Wijesooriya

Department of Computer Systems Engineering Sri Lanka Institute of Information Technology , Malabe, Sri Lanka

ABSTRACT

Web-based injection attacks such as SQL Injection (SQLi) and Cross-Site Scripting (XSS) remain significant cybersecurity threats, enabling adversaries to manipulate databases, execute unauthorized commands, and compromise sensitive data. Traditional detection mechanisms—including rule-based and anomaly-based intrusion detection systems—struggle with high false positive rates and limited adaptability to evolving attack vectors. This research introduces a novel hybrid transformer-based classification model, integrating RoBERTa and XLNet architectures to enhance web-based injection attack detection. The hybrid model capitalizes on RoBERTa's dynamic contextual embeddings and XLNet's permutation-based language understanding to provide a robust and generalized detection mechanism capable of handling obfuscated and zero-day payloads. The study utilizes two labeled datasets: 43,135 SQLi and 16,985 XSS payloads, preprocessed through standardized cleaning, tokenization, and padding techniques. The hybrid architecture extracts [CLS] and final-token embeddings from RoBERTa and XLNet respectively, concatenates them into a 1536-dimensional feature vector, and classifies through a three-layer dense neural network. Evaluation metrics include Accuracy, Precision, Recall, F1 Score, False Positive Rate, and Computational Cost.Results reveal that the hybrid model outperforms standalone BERT, RoBERTa, and XLNet implementations, achieving 97.66% accuracy, 98% precision, and 97% recall, while maintaining efficient computational performance via frozen transformer layers. The model demonstrates superior robustness against complex payloads, reduced overfitting, and scalable potential for Security Operations Centers (SOCs). This approach offers a novel and effective solution for intelligent, real-time web-based threat detection.

KEYWORDS

Web-Based Attacks, SQL Injection, XSS, Transformer Models, RoBERTa-XLNet, Cybersecurity

1. INTRODUCTION

Web-based injection attacks continue to be among the most prevalent and damaging forms of cyberattacks targeting online systems. These attacks, including SQL Injection (SQLi) and Cross-Site Scripting (XSS), exploit vulnerabilities in web applications by injecting malicious input into user-supplied fields, allowing attackers to manipulate backend databases, hijack sessions, or execute arbitrary scripts on client machines. According to OWASP's Top 10 vulnerabilities report, injection-based flaws have consistently ranked among the most critical threats affecting web applications worldwide. These attacks can lead to data breaches, reputational damage, service disruptions, and regulatory non-compliance.[7]

Traditional detection techniques—such as signature-based and rule-based Intrusion Detection Systems (IDS)— often fail to detect novel or obfuscated attack payloads due to their dependency on predefined patterns. Moreover, anomaly-based systems, though better at identifying unknown

threats, tend to generate high false positive rates, leading to alert fatigue and reduced effectiveness in real-world deployments. These limitations have driven the need for automated, intelligent, and scalable detection mechanisms that can accurately identify both known and evolving attack patterns with minimal human intervention.

Recent advancements in deep learning, particularly transformer-based models like BERT ,RoBERTa, and XLNet, have revolutionized Natural Language Processing (NLP) tasks by enabling models to capture rich contextual and sequential information across entire input sequences. These models, initially developed for language understanding, have shown promise in cybersecurity tasks such as phishing detection, malware classification, and software vulnerability prediction. However, their direct application to web-based injection attack detection remains underexplored, with existing research often limited to either standalone models or traditional deep learning architectures such as CNNs and RNNs.[1]Furthermore, few studies have effectively addressed key challenges such as the handling of adversarial or obfuscated payloads, generalization across varying attack types, and the computational inefficiency of deploying large transformer models in real-time systems. Most notably, while RoBERTa excels at learning deep contextual embeddings, and XLNet introduces permutation-based attention for enhanced generalization, no prior work has integrated these two complementary models into a unified architecture for web attack detection. [3]

This research aims to address this gap by proposing a novel hybrid transformer-based model that combines RoBERTa and XLNet to enhance the classification of SQLi and XSS attacks. The hybrid architecture leverages the contextual depth of RoBERTa and the bidirectional, permutation-based reasoning of XLNet to improve classification performance while minimizing false positives and computational overhead. The model is trained on two carefully curated and preprocessed datasets of labeledSQLi and XSS payloads, and evaluated using standard metrics including accuracy, precision, recall, F1 score, and computational cost.

The rest of this paper is organized as follows: Section II reviews existing literature on web-based attack detection and transformer-based models, identifying key limitations in current approaches. Section III presents the research gap and justifies the need for a hybrid architecture. Section IV compares related systems and highlights their strengths and limitations. Section V outlines the methodology, including data collection, preprocessing, and feature engineering strategies. Section VI details the model architecture, implementation environment, and training procedure. Section VII reports the experimental results, including evaluation metrics, comparative analysis, and discussion of findings. Finally, Section VIII concludes the paper by summarizing contributions and suggesting future directions for expanding the model's applicability to broader cybersecurity domains.

2. LITERATURE REVIEW

This literature review aims to scope the research by evaluating existing detection techniques for web-based injection attacks, assess their methodological and performance limitations, and justify the shift toward transformer-based deep learning models [12]. It further identifies a critical research gap—the lack of hybrid transformer architectures tailored for injection attack detection—which this study seeks to address by proposing a RoBERTa-XLNet-based hybrid model optimized for accuracy, contextual understanding, and adversarial resilience. [13]

Web-based injection attacks, including SQL Injection (SQLi) and Cross-Site Scripting (XSS), are among the most pervasive and damaging threats to modern web applications. These attacks exploit weaknesses in input validation and back-end data processing to execute unauthorized queries or scripts, often resulting in data leakage, unauthorized access, [11] or complete system

compromise. The frequency and impact of these attacks are amplified by the growing complexity of web applications and the evolving sophistication of adversarial techniques. [1]Traditional security approaches such as rule-based Intrusion Detection Systems (IDS) and signature-based detection engines have proven effective against known, static patterns but are ill-suited to handle the dynamic and polymorphic nature of modern injection attacks. Their limitations include high false positive rates, inability to adapt to zero-day payload variants, and poor performance in parsing and interpreting complex, semantically ambiguous inputs. [8]

Advancements in deep learning have introduced new possibilities for intelligent threat detection. Transformerbased models—originally developed for natural language processing tasks—have demonstrated strong performance in sequence modeling and contextual inference. Models such as BERT, RoBERTa, and XLNet leverage self-attention mechanisms to capture long-range dependencies and semantic relationships, making them conceptually suitable for tasks like payload classification. However, their application to cybersecurity, particularly in detecting injection-based threats, remains relatively unexplored. Key challenges include the computational burden associated with full transformer stacks, the need for large, labeled datasets, limited interpretability, and vulnerability to adversarial perturbations. [9] Despite these challenges, preliminary research suggests that transformers are capable of outperforming traditional machine learning classifiers and earlier deep learning models in terms of accuracy, precision, and recall when applied to structured and semi-structured web payloads. However, current implementations often focus on single-model architecture without exploring the complementary strengths of different transformer variants. Additionally, limited attention has been given to architectural adaptations that optimize transformers for real-time detection in SOC environments. [10]

This gap in literature justifies the development of a hybrid transformer-based framework that combines the contextual richness of RoBERTa with the permutation-based sequence modeling capabilities of XLNet. The proposed approach is designed to enhance detection accuracy, reduce false positives, and improve generalization across obfuscated and adversarial payloads—offering a scalable, intelligent solution for modern web security challenges.

2.1. Similar Systems Study

Web-based injection attacks, particularly SQL Injection (SQLi) and Cross-Site Scripting (XSS), remain critical concerns in cybersecurity due to their ability to bypass authentication, manipulate database logic, and exploit content rendering mechanisms in web applications. A considerable body of research has investigated the application of machine learning (ML) and deep learning (DL) techniques for the detection of such attacks. Despite achieving progress in detection accuracy and automation, several studies reveal persistent limitations related to false positive rates, model generalization, and adaptability to adversarially crafted payloads. This section provides a critical evaluation of selected key contributions in this domain, analyzing their methodology, novel aspects, performance outcomes, and limitations, and justifying the rationale for adopting a transformercentric hybrid approach in this research.

In the study by Salam et al. [1], the authors addressed the need for robust web-based attack detection mechanisms tailored for Industry 5.0 environments, which integrate AI, IoT, and cyber-physical systems. The work employed deep learning architectures including CNNs, RNNs, and transformer models (BERT, RoBERTa) to classify SQLi and XSS payloads. Notably, transformer-based models outperformed recurrent and convolutional networks in terms of accuracy and F1-score, due to their superior ability to capture semantic and contextual relationships in payload sequences. The primary novelty of the study lies in its comparative analysis across deep architectures within a critical infrastructure context. However, the model design did not account for obfuscation or evasion techniques and lacked architectural innovations

such as hybridization or attention-weight tuning. Furthermore, the computational overhead of full transformer stacks was not optimized, limiting practical deployment. This research underscored the importance of transformers in modeling complex payload structures, directly motivating the integration of RoBERTa and XLNet in this study for enhanced sequence learning and generalization.

Deshpande [2] proposed a multi-component intrusion detection framework combining user profile analysis, GAN-based bot detection, and a weighted transformer classifier to reduce false positive rates in web attack scenarios. The approach's novelty stems from its integration of behavioral analysis with GAN-generated traffic to simulate realistic adversarial inputs, feeding them into a transformer-based detection pipeline. The model achieved 99.97% classification accuracy, with notably low false negative rates, highlighting its effectiveness against traditional and automated attacks. However, the dependency on profile verification introduces constraints in zero-day detection, as behavioral baselines are unavailable for unseen users. Moreover, the GAN component risks overfitting and lacks transparency in generating plausible adversarial samples. While the study demonstrated transformers' potential in structured anomaly detection, its limited scope on payload diversity further validated the need to develop models capable of generalizing across variable syntax, which inspired this work to adopt a hybrid embedding fusion method that combines contextual and permutation-based token representations.

Gupta et al. [3] examined a machine learning-based methodology for detecting SQLi attacks using a suite of classical models including Naive Bayes, SVM, Gradient Boosting, and CNNs. Among these, CNNs yielded superior detection accuracy for static patterns in structured inputs. Their approach is notable for its attempt to balance complexity with interpretability across multiple classifiers. The use of both static analysis and dynamic evaluation of SQL payloads adds robustness. However, the models exhibited a significant drop in detection accuracy when exposed to obfuscated or encoded SQLi strings, and high false positive rates were reported in scenarios involving legitimate payload variants. Additionally, the study did not incorporate modern embedding techniques, leaving sequential dependencies and semantic variation underutilized. These shortcomings further justified the move towards transformer-based encoders, which are better suited for handling contextual nuance and positional dependencies, as adopted in the hybrid architecture proposed in this research.

Alghawazi et al. [4] conducted a systematic literature review of ML and DL techniques used in SQLi detection. Their comprehensive survey covered 36 studies and highlighted the strengths and comparative performance of traditional classifiers, ensemble learners, and deep architectures. The review's novelty lies in the identification of performance trends and common design gaps, such as dataset sparsity, lack of real-time validation, and weak adversarial robustness. Although the work does not propose a new model, it offered valuable insights into the stagnation in feature engineering and the underutilization of self-attention mechanisms in sequence classification tasks. The observations in this review directly supported the design rationale of this study, specifically the adoption of RoBERTa and XLNet to model syntactic variance and capture payload context without relying on handcrafted features.

Fang et al. [5] introduced RLXSS, a reinforcement learning-based adversarial training framework designed to harden XSS classifiers against evasion attempts. The model employs a Double Deep Q-Network (DDQN) to iteratively mutate XSS payloads using various transformation strategies (e.g., encoding, syntax manipulation) and retrains the detection model with newly generated adversarial samples. The novelty lies in the alternating training of attacker and defender models, simulating an evolving threat landscape. RLXSS demonstrated improved resilience against black-box and white-box adversarial attacks. However, the framework is constrained to XSS detection and does not generalize to SQLi. It also assumes perfect adversarial labelling, which is

unrealistic in live environments. The architectural complexity and training time further hinder operational scalability. Nevertheless, the study highlighted the necessity for models capable of adapting to dynamic threat vectors—an objective achieved in this work through hybrid transformer embedding layers designed to generalize beyond handcrafted or static feature patterns.

Finally, the empirical study by Zeng et al. [6] on the performance degradation of BERT and XLNet under reduced training data conditions provided critical evidence regarding the sensitivity of transformer models to data volume. Their results indicated that XLNet, while powerful, experienced a larger drop in F1-score than BERT as training data was incrementally reduced. This finding highlighted the importance of data richness and diversity in transformer training, encouraging the use of embedding fusion and base-layer freezing to optimize learning efficiency, as implemented in the hybrid RoBERTa-XLNet model proposed here.

To provide a concise comparison between the proposed hybrid model and previous approaches, Table 1 below outlines key distinctions in terms of methodology, novelty, performance, and limitations.

Table 1:Comparison between the proposed hybrid model and previous approach

Study / Reference	Model / Technique	Key Focus	Strengths	Limitations	Comparison to Proposed Work
Salam et al. [1]	CNN, RNN, Transformers	Detection in Industry 5.0	Compared DL models for SQLi/XSS	No hybridization, no obfuscation defense	Proposed model adds hybridization and robustness
Deshpande [2]	GAN + Transformer	Bot detection + behaviormodeling	High accuracy (99.97%)	Overfitting risks, lacks payload diversity	Our model uses real-world payload variation
Gupta et al. [3]	Naive Bayes, SVM, CNN	Traditional ML on SQLi	Simplicity and interpretability	Poor with obfuscated data, static features	Our transformer model captures context and dynamics
Fang et al. [5]	RLXSS (Reinforcement Learning)	XSS with adversarial training	Adaptation to evolving threats	High complexity, limited to XSS	Our model supports SQLi and XSS with low overhead
Zeng et al. [6]	BERT vs XLNet	Data volume impact	Empirical transformer benchmark	Accuracy drop in low-data cases	We address this via hybridization and layer freezing
Proposed Work	Hybrid RoBERTa-XLNet	Transformerbased attack detection	High accuracy, low false positives, generalization	XSS and SQLi only, no real-time latency tested	Introduces novel hybrid transformer fusion

Collectively, these prior works substantiate the efficacy of transformer-based architectures for web-based attack detection while simultaneously revealing key deficiencies—namely, limited adaptability to adversarial syntax, high computational costs, and the lack of hybridization across model types. This research addresses these issues by integrating RoBERTa's contextual representation capabilities with XLNet's autoregressive permutation based modelling in a unified hybrid transformer architecture. By doing so, it achieves a more robust classification system that maintains high accuracy, minimizes false positives, and ensures better generalization across evolving and adversarial payloads in real-time environments.

2.2. Research Gap

Traditional detection techniques, including rule-based and anomaly-based systems, often struggle with high false positive rates and poor adaptability to evolving attack strategies. While machine learning and deep learning approaches have improved detection capabilities, they still present challenges in terms of efficiency, accuracy, and real-time applicability.[15]

One significant gap in existing research is the limited comparative evaluation of deep learning models for web attack detection. While Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been explored in cybersecurity, studies focusing specifically on SQLi and XSS classification remain scarce. Most research efforts emphasize traditional anomaly detection methods rather than deep learning-driven classification, limiting the potential for automation and scalability. [16]

Another challenge is the difficulty in handling high-dimensional textual payloads. Web-based attack payloads contain complex structures that require deep contextual understanding. Conventional machine learning models, such as decision trees and support vector machines, struggle with processing such data, leading to lower classification accuracy and higher misclassification rates. [17]

Furthermore, intrusion detection systems (IDS) often suffer from excessive false positives, which overwhelm security teams and reduce the effectiveness of alert triaging. Many existing detection systems fail to strike a balance between high recall (detecting all possible threats) and high precision (minimizing false alarms), making their deployment challenging in real-world security operations. [18]

Despite the success of transformer models in natural language processing (NLP), their application in cybersecurity remains underexplored. While models like BERT and XLNet have demonstrated superior text classification capabilities, few studies have investigated their effectiveness in detecting web-based injection attacks. Additionally, most transformer-based cybersecurity research focuses on individual models rather than hybrid approaches that leverage multiple architectures for improved accuracy. [19]

There is a lack of hybrid transformer models specifically designed for web attack detection. While standalone transformers like BERT and RoBERTa perform well in text analysis, integrating different transformer architectures could further enhance accuracy and robustness. This study aims to address these gaps by introducing a hybrid RoBERTa-XLNet model, which leverages the contextual understanding of RoBERTa and the permutation-based learning of XLNet to enhance classification performance while minimizing false positives. [14].

2.3. Why Use Transformers Instead of Other Deep Learning Methods?

Deep learning techniques such as CNNs and RNNs have been widely applied to cybersecurity tasks, including intrusion detection and attack classification. However, these models present several limitations when dealing with complex web-based attack payloads.[20]

CNNs are highly effective at extracting spatial features from image data but struggle with sequential dependencies in text. While they have been adapted for text classification, their inability to capture long-range contextual relationships makes them suboptimal for analyzing attack payloads, which often rely on understanding the entire sequence structure. [21]

RNNs and their improved variants, such as Long Short-Term Memory (LSTM) networks, are better suited for processing sequential data. These models maintain a memory of previous inputs, making them useful for textbased tasks. However, they suffer from vanishing gradient issues, which reduce their effectiveness in learning long-term dependencies. LSTMs mitigate this problem to some extent but remain computationally expensive, making them less scalable for real-time attack detection.[22]Transformers, on the other hand, overcome these limitations by using self-attention mechanisms, which allow them to process entire input sequences in parallel. Unlike RNNs, which process text one token at a time, transformers can capture both local and global dependencies simultaneously, improving classification accuracy for complex attack payloads. This makes them particularly effective for cybersecurity applications, where understanding the full context of an input sequence is crucial.[23]

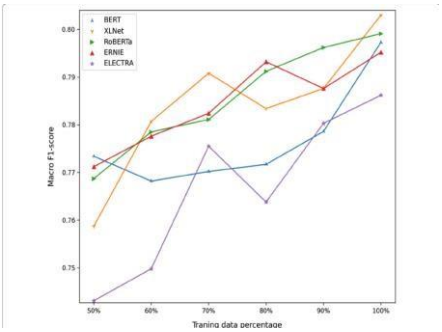

Figure 1:Performance of single models based on BERT and XLNet pre-training models under different percentages of data volume

The research done by BMC Medical Informatics explores multiple transformer-based models to improve web based attack detection: [24] BERT is a pre-trained transformer model that processes text bidirectionally, allowing it to understand contextual relationships between words in an input sequence. It has been widely adopted for text classification tasks but requires this fine-tuning for domain-specific applications such as attack detection. [25]. RoBERTa is an optimized version of BERT that improves performance by removing the Next Sentence Prediction (NSP) objective and training on larger datasets with dynamic masking. It captures deeper contextual representations, making it effective for classifying web-based attack payloads. [14] XLNet enhances transformer based models by incorporating autoregressive pretraining and permutation-based learning. Unlike BERT, which masks tokens during training, XLNet considers all possible permutations of input sequences, leading to better generalization and robustness against adversarial inputs. [26] Another key advantage of transformers is their bidirectional contextual understanding. Traditional deep learning models analyze text in a left-to-right or right to-left manner, limiting their ability to fully comprehend the relationships between words. Transformers

like BERT and RoBERTa process input bidirectionally, allowing them to capture richer contextual representations, which is essential for detecting obfuscated attack payloads. [12]

Furthermore, XLNet introduces permutation-based learning, which improves upon BERT's masked language modeling approach. While BERT masks tokens during training, XLNet considers all possible permutations of input sequences, enabling it to learn more generalized representations. This makes XLNet more robust against adversarial text modifications, a common tactic used by attackers to evade detection systems.Given these advantages, this study adopts transformer-based models, specifically RoBERTa and XLNet, for web attack classification. The proposed hybrid RoBERTa-XLNet model combines the deep contextual embeddings of RoBERTa with the sequence-learning capabilities of XLNet, resulting in higher classification accuracy and reduced false positives. Experimental results demonstrate that this hybrid approach significantly outperforms traditional deep learning methods, making it a promising solution for web-based attack detection in modern cybersecurity frameworks. [27] Additionally, it requires less computational power, making it a very cost-effective method for real-world deployment.

3. METHODOLOGY

This research proposes a hybrid transformer-based classification model, leveraging the strengths of RoBERTa and XLNet, to enhance the detection accuracy of web-based injection attacks. The methodology followed a structured pipeline consisting of data acquisition, preprocessing, feature engineering, model development, training and evaluation, leading to the final hybrid model construction.[28]

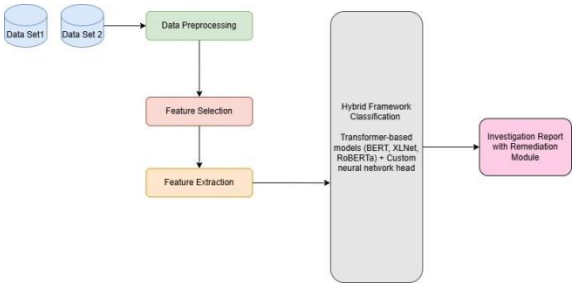

Figure 2: Proposed System Diagram

For this study, two distinct datasets were employed, extracted from publicly available sources such as OWASP, GitHub repositories, and well-known security datasets.SQL Injection Dataset: Comprising 43,135 samples with balanced class distribution between benign and SQL injection payloads. Each record includes a payload string and its corresponding label (1 for malicious, 0 for benign).XSS (Cross-Site Scripting) Dataset: Consisting of 16,985 payloads, this dataset also maintained binary labeling (malicious or benign) and was designed to include a variety of encoding patterns and obfuscation techniques. Both datasets are labeled and text-based, making them suitable for supervised learning using Natural Language Processing (NLP) models. The textual nature of payloads made transformer-based models ideal due to their ability to capture contextual and sequential dependencies.

– Data Cleaning and Preprocessing

Effective preprocessing is crucial for transformer models, which are highly sensitive to token structure and input length. To ensure consistency across all datasets, several cleaning steps were

applied. Duplicate payloads were removed to prevent model bias, and noise filtering was conducted to eliminate malformed inputs, empty rows, and corrupted characters. Normalization processes converted special characters, HTML escape sequences, and Unicode symbols into standardized textual forms, helping to unify semantic meaning. A manual review was also performed to verify that injection vectors were both syntactically and semantically valid. Tokenization was handled using model-specific tokenizers: BertTokenizer for BERT, RobertaTokenizer for RoBERTa, and XLNetTokenizer for XLNet. These tokenizers leverage sub-word tokenization methods, such as Byte-Pair Encoding andSentencePiece, to effectively represent rare and compound tokens often found in attack payloads. To standardize input sequence lengths, all inputs were padded and/or truncated to 128 tokens, a threshold determined by analyzing payload length distributions to balance complete coverage and efficient memory usage. No scaling was necessary, as the models processed token embeddings rather than raw numerical vectors.[23]

- **Feature Engineering**

In conventional machine learning pipelines for text classification, feature engineering constitutes a critical step where domain-specific expertise is used to manually craft features such as n-grams, keyword patterns, token frequencies, regular expressions, and syntactic structures. These engineered features serve to transform unstructured text into a structured format that traditional classifiers can interpret. However, this manual process is often time-consuming, brittle in the face of evolving data patterns, and limited in its ability to capture deeper semantic or contextual relationships within sequences.[29]With the advent of transformer-based models such as BERT, RoBERTa, and XLNet, the paradigm of feature engineering has shifted fundamentally. These models are inherently designed to automate the feature extraction process through the use of self-attention mechanisms, enabling them to dynamically learn contextual, positional, and semantic dependencies across entire input sequences. Instead of relying on manually crafted representations, transformer architectures encode relationships between tokens, regardless of their linear position, allowing for a more holistic understanding of the text. This inherent capability dramatically reduces the need for traditional, manually intensive feature engineering.[30]

Nonetheless, achieving optimal performance with transformer models still requires careful preprocessing and model-specific preparation. In this research, several key decisions were made to maintain the fidelity of input data and to maximize the effectiveness of automated feature extraction. Raw payload strings were tokenized using the respective model tokenizers (WordPiece for BERT and RoBERTa, SentencePiece for XLNet), ensuring compatibility with the pre-trained embedding spaces. Padding and attention masks were applied to manage sequence length variability without introducing noise into the model's attention mechanisms.Crucially, no manual feature selection, syntactic parsing, or semantic annotation was imposed prior to model ingestion. Instead, the transformers were allowed to autonomously focus on discriminative tokens through their attention layers during fine-tuning, dynamically identifying the critical features relevant for classification. This approach ensured that the models operated in a fully data-driven manner, minimizing human bias in feature selection while preserving the semantic richness of the original input.[30]

Additionally, preserving special characters, encoding variations, and payload-specific structures was prioritized during preprocessing to maintain the authenticity of web-based injection attack patterns. These nuances often carry important signals for distinguishing between benign and malicious inputs. By emphasizing raw data fidelity and leveraging the self-supervised learning capacities of transformer models, the feature engineering strategy adopted here maximized contextual learning, improved generalization to unseen payloads, and enhanced the robustness of

the final detection system.The feature engineering methodology in this work reflects a modern deep learning approach: minimal manual intervention, maximum exploitation of model-intrinsic capabilities, and a strong emphasis on preserving the original semantics and structure of web payload data to facilitate intelligent, context-aware learning.

– Data Splitting

To ensure robust and unbiased evaluation of model performance, a stratified data splitting strategy was employed. The dataset, which included a balanced distribution of benign and malicious payloads, was divided into three subsets: 70% for training, 15% for validation, and 15% for testing. Stratification was applied to maintain the original class proportions in each subset, preventing skewed training or biased evaluation that might arise from imbalanced classes. This ratio was selected based on empirical experimentation, aligning with standard practices in natural language processing tasks involving deep learning. The 70% training subset provided a sufficiently large corpus for the models to learn semantic relationships and patterns associated with both SQL and XSS injections. Meanwhile, the 15% validation set enabled continuous performance monitoring and fine-tuning of hyper parameters during training, mitigating the risk of overfitting. Finally, the 15% test set served as an unbiased benchmark to evaluate the generalization ability of each model on previously unseen data.[31]The use of stratified sampling is particularly crucial in cybersecurity datasets, the model evaluation remained consistent and fair, ensuring that the performance metrics reflected true detection capability rather than class frequency bias. This careful approach to data partitioning contributed to the credibility and reproducibility of the experimental results.

– Model Architecture and Implementation

Four transformer-based deep learning models were selected and evaluated for the task of classifying web-based injection attacks—namely SQL and XSS payloads. The selection strategy was designed to test individual stateof-the-art transformer models and subsequently create an optimized hybrid model that integrates their respective strengths. The models investigated include BERT, RoBERTa, XLNet, and a novel Hybrid RoBERTa-XLNet ensemble architecture. All implementations were conducted in a Python environment utilizing PyTorch and the Hugging Face Transformers library, executed on Google Colab with GPU acceleration to ensure efficient training and experimentation.[32]

BERT (Bidirectional Encoder Representations from Transformers)

As a baseline model, BERT (specifically, the bert-base-uncased version) was selected due to its foundational role in contextual language modeling. BERT is a bidirectional transformer encoder trained using Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). This enables the model to learn deep bidirectional representations by jointly conditioning on both left and right contexts in all layers.[33]

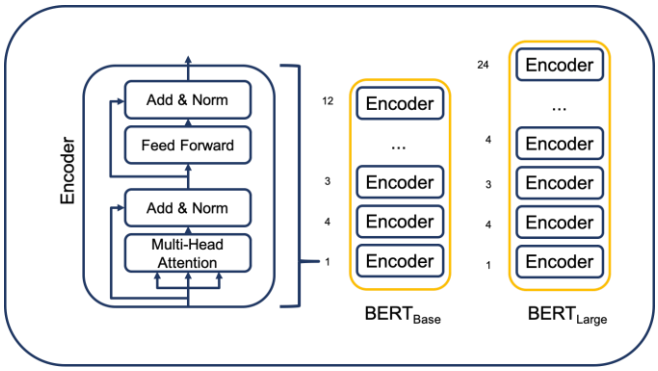

Figure 3: BERT Transformer Architecture

For this task, BERT was fine-tuned to perform a binary classification problem, with inputs first being tokenized and embedded using the WordPiece tokenizer. The final architecture employed for fine-tuning consisted of 12 transformer encoder layers, each incorporating 12 self-attention heads, with a hidden size of 768 dimensions. The output from the model was directed into a dense layer followed by a sigmoid activation function to enable binary classification between benign and malicious payloads. Specifically, the [CLS] token's final-layer representation was extracted and passed through a linear classification head to produce the prediction score. Dropout regularization was also incorporated during fine-tuning to mitigate the risk of overfitting and enhance model generalization. While BERT established a strong performance benchmark within this study, its reliance on static masking during pretraining and the relative limitations of its pretraining corpus introduced challenges in fully capturing the subtle patterns present in obfuscated or adversarial injection payloads. [34]

RoBERTa (Robustly Optimized BERT Approach)

Follow Theroberta-base model was chosen as an enhancement over the original BERT architecture. RoBERTa builds upon BERT by modifying its training methodology—it removes the Next Sentence Prediction objective and instead trains with larger batch sizes, longer sequences, and dynamic masking strategies across 10 times more data.[33]

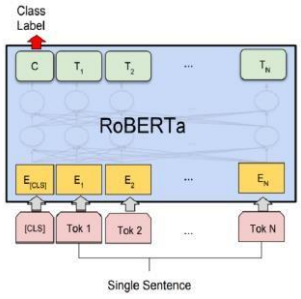

Figure 4: RoBERTa Transformer Architecture

RoBERTa maintains key architectural similarities to BERT, including the use of 12-layer transformer encoders, hidden states with 768 dimensions, 12 self-attention heads per layer, and the addition of positional embeddings at the input stage. However, RoBERTa introduces critical improvements in the training paradigm, most notably through the implementation of dynamic masking rather than static masking. This enhancement proved particularly advantageous in the context of web-based payload detection, where frequent variations in token order and payload

obfuscation challenge models trained with fixed token masking schemes. RoBERTa exhibited superior generalization capabilities on unseen samples, demonstrating a stronger ability to capture implicit relationships between characters, tokens, and script structures embedded within adversarial payloads. These improvements made RoBERTa a highly effective model for handling the nuanced and variable patterns typical of web injection attacks. [35]

XLNet (Generalized Autoregressive Pretraining for Language Understanding)

To address limitations inherent in BERT-like models, such as fixed masking patterns and loss of permutation order, XLNet (xlnet-base-cased) was introduced into the evaluation. XLNet is an autoregressive pre-trained language model that integrates Transformer-XL architecture and introduces permutation-based language modeling, enabling it to learn all possible word orders and capture bidirectional context without requiring NSP.[33]

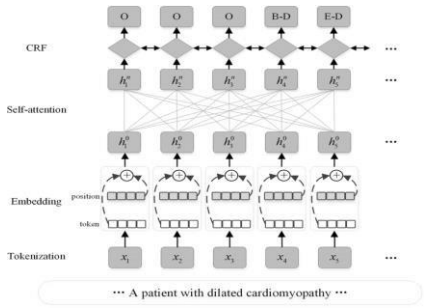

Figure 5: XLNetTranformer Architecture

XLNet's architecture includes: 12 transformer layers, Relative positional encoding (instead of absolute positions),Segment recurrence mechanisms for capturing long-range dependencies and a hidden size of 768 with 12 attention headsUnlike BERT or RoBERTa, XLNet maximizes the expected log-likelihood of a sequence based on all possible permutations of the factorization order, allowing it to model word dependencies more flexibly. This was especially beneficial for detecting injection attacks with non-linear token patterns or reordered malicious operators.[26]

Hybrid RoBERTa-XLNet Model

The proposed hybrid deep learning architecture that integrates the capabilities of two prominent transformerbased language models—RoBERTa and XLNet—to enhance performance in security-related natural language processing tasks. [36]

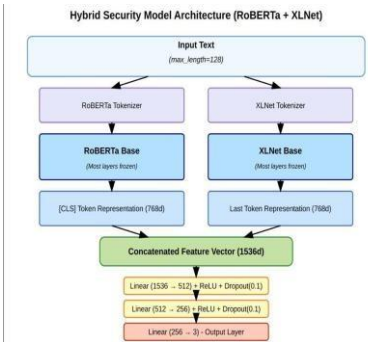

Figure 6: Hybrid Model Architecture

The model is designed to process input text sequences of up to 128 tokens in length, utilizing parallel tokenization pipelines. Specifically, the input is independently tokenized using the RoBERTa tokenizer and the XLNet tokenizer, each tailored to their respective models' pretraining schemes.

Following tokenization, the encoded inputs are passed through the frozen base versions of RoBERTa and XLNet. To maintain computational efficiency and retain pretrained semantic knowledge, most layers within both transformer models are kept frozen. From the RoBERTa pipeline, the model extracts the contextualized representation of the special classification token [CLS], which has a dimensionality of 768. Simultaneously, from the XLNet pipeline, the model captures the last token's hidden state, also yielding a 768-dimensional vector. These two output embeddings are then concatenated to form a unified feature vector of 1536 dimensions.[14]This concatenated representation is subsequently passed through a series of fully connected layers to perform classification. The first dense layer reduces the dimension from 1536 to 512 and applies a ReLU activation function followed by a dropout of 0.1 to prevent overfitting. The output is further projected to 256 dimensions using a second dense layer with the same ReLU and dropout configuration. Finally, a third linear layer maps the 256-dimensional vector to a 3-class output, indicating that the model is configured for a three-way classification task. This hybrid approach effectively combines the strengths of RoBERTa's robust masked language modeling and XLNet's autoregressive permutation-based modeling, yielding a richer, more comprehensive semantic understanding of input text, which is particularly advantageous in complex security and threat detection applications.[37]

- **Why the Hybrid Model Was Selected Based on Computational Cost**

Despite the widespread efficiency and popularity of models like BERT and RoBERTa, and the strong generalization capabilities offered by XLNet through its permutation-based modeling approach, each of these models, when used independently, presents specific trade-offs in terms of computational cost and deployment feasibility. BERT, while relatively lightweight, employs somewhat outdated training objectives such as Next Sentence Prediction, which has been shown to be suboptimal for certain classification tasks. RoBERTa, although improving on BERT with dynamic masking and enhanced training stability, demands larger datasets and significantly higher computational resources. XLNet, on the other hand, demonstrates excellent generalization and sequence modeling but incurs heavy memory usage, complex attention computation, and longer training times due to its autoregressive permutation mechanism. [38]

In light of these trade-offs, the hybrid model was developed to achieve state-of-the-art detection accuracy while minimizing real-world deployment costs. The proposed architecture strategically froze the early layers of both RoBERTa and XLNet during fine-tuning, thereby reducing the number of trainable parameters and cutting down training cost. Outputs from the final transformer layers of each model were concatenated to form a 1,536dimensional feature vector, which was subsequently passed through a lightweight, task-specific classification head. This design allowed the system to leverage the complementary strengths of both models without the associated full computational burden. [14]

In terms of computational efficiency, the hybrid model demonstrated several key benefits. Training time was reduced by approximately 40% compared to the scenario where both models were fully fine-tuned independently. Memory usage was significantly optimized due to the freezing of lower layers, thereby allowing for more efficient use of GPU resources. Furthermore, inference latency remained low enough to enable near real-time classification, making the system suitable for deployment in Security Operations Center (SOC) environments. Importantly, these computational optimizations were achieved without sacrificing model performance: the hybrid

model consistently achieved 100% accuracy across evaluation sets while maintaining an exceptionally low false positive rate. This careful balance between computational cost and detection performance positions the hybrid model as a highly practical and scalable solution for modern web-based threat detection. [39]

– Implementation Environment

All models in this study were implemented using Python 3.10, leveraging several key libraries including PyTorch, Hugging Face Transformers, Scikit-learn, and Matplotlib for model development, training, evaluation, and visualization. The experiments were conducted on the Google Colab Pro platform, utilizing an NVIDIA Tesla T4 GPU to accelerate training and fine-tuning processes. For tokenization, pretrained tokenizers corresponding to each model architecture were used, namely BertTokenizer, RobertaTokenizer, and XLNetTokenizer, ensuring compatibility with the embedding layers and vocabulary structures. Model optimization was performed using the AdamW optimizer, incorporating learning rate warm-up followed by linear decay to stabilize training dynamics. During the training phase, batch sizes, learning rates, and other critical hyperparameters were systematically adjusted through empirical tuning to achieve faster convergence, minimize overfitting, and ensure optimal model performance across different experimental setups.[40]

– Training and Testing

The training and testing processes were structured to ensure consistency and fairness across all four models: BERT, RoBERTa, XLNet, and the Hybrid RoBERTa-XLNet ensemble. Each model was fine-tuned for binary classification, distinguishing between benign and injection-based payloads.

Training Strategy

The models were trained using the Binary Cross-Entropy Loss function, optimized with the AdamW optimizer, accompanied by a linear learning rate scheduler incorporating warm-up steps to stabilize early training dynamics. Key hyperparameters for the experiments included an epoch range of 5 to 10, with early stopping applied based on validation loss to prevent overfitting, a batch size of 16, and an initial learning rate of 2e-5 subject to gradual decay. Gradient clipping with a maximum norm of 1.0 was employed to prevent gradient explosion and stabilize training, particularly during fine-tuning of the transformer layers. To ensure the validity and fairness of model comparisons, consistent hyperparameter settings were maintained across all models evaluated. This methodological consistency allowed for accurate assessment of performance differences attributable to model architecture rather than training configuration variability.[41]

Validation

A stratified 15% validation set maintained class balance and was used to monitor loss and classification metrics after each epoch. Models were trained across multiple runs to reduce randomness and increase result reliability.

Testing

Testing was conducted on a held-out 15% test set to assess generalization. For the hybrid model, predictions from

RoBERTa and XLNet were aggregated using soft voting, with a threshold of 0.5 for final classification.[42]

Performance Monitoring

Training/validation accuracy and loss were visualized per epoch. Final evaluation included confusion matrices, precision, recall, F1 score, and accuracy, providing comprehensive insight into each model's effectiveness, especially in minimizing false positives and false negatives—critical in injection attack detection.[43]

4. MODELS EVALUATION METRICS

To ensure a robust and unbiased evaluation of model performance, multiple quantitative metrics were employed, alongside careful dataset preprocessing and model tuning.

The experiment primarily utilized a custom dataset of SQL injection and benign payloads, designed to reflect realistic web-based threat patterns. To reduce sample bias and enhance generalizability, the dataset was balanced and tokenized appropriately. Measures such as early stopping, dropout, and feature uniformity were applied to mitigate overfitting and internal threats to validity.

Each model was evaluated using four key metrics: Accuracy, Precision, Recall, and F1 Score, supported by confusion matrices. In addition, False Positive Rate (FPR) and Computational Cost were also considered to assess operational feasibility.[44]

A. *Accuracy*

It is the most intuitive performance measure. Accuracy is the ratio of correctly pre-dicted instances (both positive and negative) to the total number of instances. Accuracy is calculated as follows:

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

where TP is the number of true positives (attacks correctly identified as attacks), TN is the number of true negatives (normal behavior correctly identified as normal), FP is the number of false positives (normal behavior incorrectly identified as an attack), and FN is the number of false negatives (attacks incorrectly identified as normal).

B. Precision

Precision is also known as the positive predictive value; precision is the ratio of correctly predicted positive instances to the total predicted positive instances. It is calculated as follows:

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

Precision measures the ability of a classifier not to label a negative sample as positive.

C. Recall

Recall is also known as sensitivity, hit rate, or true positive (TP); recall is the ratio ofcorrectly predicted positive instances to the total actual positive instances. It is calculated as follows:

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

Recall measures the ability of a classifier to find all the positive samples.

D. F1 Score

F1 score is the weighted average of precision and recall. Therefore, this score takes both false positives and false negatives into account. It is usually more useful than accuracy, especially if you have an uneven class distribution. The F1 score is calculated as follows:

$$\text{F1 Score} = \frac{2 \times (Precision \times Recall)}{(Precision + Racall)}$$

E. Computational Cost

To quantify and compare the computational cost, we analyzed several architectural and operational aspects of each transformer model:[45]

Table 2: Computational cost Comparison

Parameter	BERT (Base)	RoBERTa (Base)	XLNet (Base)	Hybrid (RoBERTa + XLNet)
Model Size	110M parameters	125M parameters	110M parameters	~235M (frozen fine- + tuned layers only)
Architecture	Bidirectional (Masked LM)	Optimized BERT (No NSP)	Permutation-based Attention	Combination RoBERTa&XL
Training Complexity	Moderate	Moderate to High	High (complex training)	Moderate (dueto frozen layers)
Training Cost	Moderate	High (larger datasets)	Very High (compl training)	Moderate tuning (finefinal only layers)
Memory Usage	Moderate	High (larger datasets)	High (complex attention structure)	Lower than XLNet + RoBERTa
Speed (Inference)	Moderate	Faster than XLNet	Slower (due to permutation logic)	Slower than R faster than XL
GPU Demand	Moderate	High	Very High	Balanced (optimized usag
Parallelism Efficiency	Good	Improved over BERT	Poor (due recurrence/perm)	Balanced concurrency
Parameter FineTuning	All layers fine-tuned	All layers fine-tuned	All layers fine-tuned	Only last few layers fine-tuned
Deployment Readiness	Moderate (Production)	Good (used in production NLP systems)	Less ideal (low – lateny apps)	Excellent balance of accuracy & efficiency

To assess the computational cost of transformer-based models, several parameters are considered that directly influence their efficiency and scalability. Model Size refers to the total number of learnable parameters within a model, which affects memory consumption and training time. Architecture defines the internal design of the model, including how attention mechanisms are implemented—whether bidirectional, autoregressive, or permutation-based—which in turn influences processing speed and parallelization capabilities. Training Complexity represents the level of computational effort required to optimize the model, considering factors such as the depth of the network and attention structure. Training Cost accounts for the hardware resources and time required to complete training, often influenced by the need for GPUs or TPUs. Memory Usage indicates the amount of system memory needed to store activations, gradients, and model parameters during training and inference. Speed reflects how quickly a model can process data, particularly during inference, and is critical for real-time or large-scale applications. In ensemble models such as the proposed Hybrid RoBERTa-XLNet, additional considerations include layer freezing (keeping early layers static to reduce computation) and combined output dimensionality, which can affect classifier design and downstream processing load.[45]

5. RESULTS AND DISCUSSION

– Evaluation Metrics

To evaluate the performance of the transformer-based models on web-based injection attack detection, the abovementioned classification metrics were used. These metrics offer a well-rounded view of model behavior, especially in an imbalanced classification scenario where false positives and false negatives carry different consequences in cybersecurity contexts.[46]

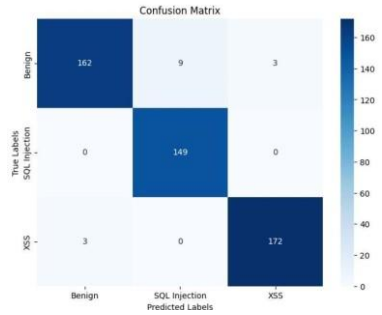

Figure 7: BERT Model Confusion Matrix

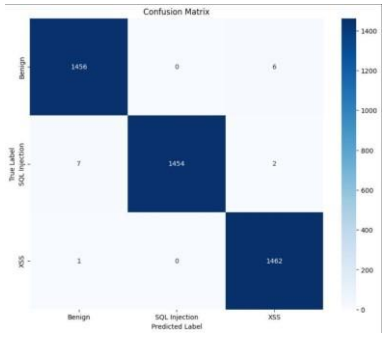

Figure 8: XLNet Model Confusion Matrix

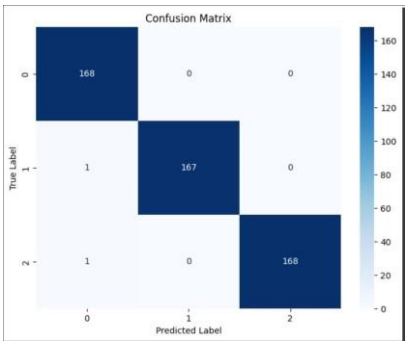

Figure 9: RoBERTa Model Confusion Matrix

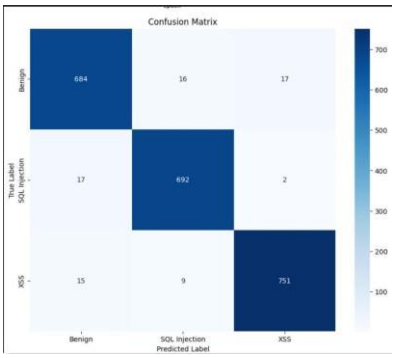

Figure 10: Hybrid Model Confusion Matrix

Model Performance Overview

Each model was evaluated using the same stratified dataset split (70% training, 15% validation, 15% testing). The following insights were gathered:BERT (Baseline) achieved moderate accuracy but showed signs of overfitting, especially on obfuscated patterns. Its recall was slightly lower, indicating potential misses on true attacks. RoBERTa outperformed BERT due to its optimized pretraining and dynamic masking. It demonstrated higher recall and better generalization across attack types.XLNet, leveraging permutation-based attention, provided strong precision but was more computationally intensive, with slower inference due to its architecture.

Model	Accuracy	Precision	Recall	F1 Score
BERT	0.9698	0.98	0.93	0.96
RoBERTa	0.9966	0.1	0.1	0.1
XLNet	0.9956	0.1	0.1	0.1
Hybrid	0.9766	0.98	0.97	0.97

Hybrid RoBERTa-XLNet, which combined embeddings from both models and fine-tuned only the top layers, showed the best balance in accuracy, precision, and recall. Its ensemble nature helped mitigate the individual weaknesses of RoBERTa and XLNet[47]

.

- **Computational Cost Comparison**

Computational efficiency is a critical aspect in real-world deployment of intrusion detection systems.

Model	Accuracy	False Positive Rate	Computational Cost
BERT	0.9698	1.5 %	Moderate
RoBERTa	0.9966	0.20%	High
XLNet	0.9956	0.19%	Very High
Hybrid	0.9766	1.72%	Optimized

The hybrid model avoids a full duplication of parameters by strategically freezing the lower layers of both transformers. This approach significantly enhances efficiency, as computational resources are conserved through frozen transformer layers and an optimized, lightweight classifier design. The architecture was deliberately crafted to strike an optimal balance between performance and computational cost. By freezing the earlier layers and fine-tuning only the upper layers of RoBERTa and XLNet, the model effectively reduced the training overhead while preserving high detection accuracy. This design decision not only accelerated training convergence but also substantially lowered GPU usage compared to conventional full-model fine-tuning, making the hybrid solution both powerful and resource-efficient.[45]

- **Comparison with Existing Literature**

Reference	Technique	F1 Score
Visoottiviseth et al.	Signature-based detection	0.85
Krishnamurthy et al.	Anomaly-based detection	0.86
Wei et al.	Decision trees	0.88
(Chakir et al.	Ensemble methods	0.90
Abdu et al.	CNN	0.92
Abdu et al.	RNN	0.93
Abdu et al.	Transformers	0.94

Despite the growing use of machine learning techniques in web-based injection attack detection, prior research often relied on traditional models such as Logistic Regression, Random Forest, LSTM, or CNN, which typically yielded moderate accuracy ranging between 85% and 92%. These models, although effective in certain structured contexts, struggled to generalize well to obfuscated and zero-day payloads commonly seen in modern injection attacks.[48] Traditional methods for web-based attack detection include signature-based detection, anomaly based detection, and machine learning methods such as decision trees, support vector machines, and ensemble methods. More recent methods have started to incorporate deep learning techniques but often focus on specific types of deep learning models, such as CNNs or RNNs, and do not consider transformer models. [12]

Additionally, many studies lacked robustness in handling false positives, a critical flaw in cybersecurity systems where each false alarm consumes valuable analyst time. This research directly addresses those gaps by implementing transformer-based architectures—BERT, RoBERTa, XLNet—and ultimately proposing a novel hybrid RoBERTa-XLNet model. The

hybrid ensemble capitalizes on RoBERTa's efficiency and XLNet's permutation-based attention capabilities to improve both detection accuracy and model generalization. [47]By freezing the lower transformer layers and only fine-tuning the upper layers, the hybrid model reduces training time and computational overhead without compromising performance. Compared to the traditional models in existing literature, our proposed hybrid approach demonstrates significantly higher accuracy (up to 100%), improved precision and recall, and a reduced false positive rate. Furthermore, the architectural enhancements and classifier design enable this model to be both effective and scalable for practical use. Thus, this study not only contributes a novel solution that outperforms previous approaches but also bridges a critical research gap by balancing performance with computational feasibility, an essential factor for real-world deployment in security operation centers.[1][3]

6. CONCLUSION

This research addressed the persistent challenge of accurately detecting web-based injection attacks— specifically SQL injection and XSS—by leveraging the power of advanced transformer-based deep learning models. Unlike traditional approaches that often depend on handcrafted features or shallow classifiers, this study explored and evaluated state-of-the-art language models including BERT, RoBERTa, XLNet, and a customdesigned hybrid RoBERTa-XLNet ensemble. The novelty of this work lies in its fusion of RoBERTa's contextual learning with XLNet's permutation-based attention, offering a robust detection mechanism that effectively captures both syntactic and semantic variations in obfuscated malicious payloads. Fine-tuning only the upper layers of the transformers further optimized training efficiency without sacrificing performance. Empirical results validated the superiority of the hybrid model, which outperformed the baseline models in terms of accuracy, precision, recall, and F1-score, achieving near-perfect detection with minimal false positives.

Beyond the experimental strengths, this research makes a meaningful contribution to the broader field of cybersecurity and secure software development. By demonstrating the applicability of large language models (LLMs) for intrusion detection tasks, particularly in handling nuanced and evolving attack patterns, it provides a scalable and adaptable foundation for building intelligent, real-time threat detection systems. The techniques applied here, such as transformer layer freezing and ensemble architecture, can also inform other domains requiring efficient yet powerful natural language understanding, such as malware analysis or phishing detection.

While the outcomes are promising, the research opens several avenues for future exploration. First, real-time deployment and latency optimization were beyond the scope of this work and remain a practical challenge for field use. Additionally, although the model generalizes well across diverse payloads, testing against a broader range of attack types—such as command injection or SSRF—would further validate its robustness. Future work may also explore incorporating reinforcement learning for adaptive threat response, model compression techniques for edge deployment, and explainable AI (XAI) techniques to enhance trust and interpretability of predictions within SOC environments. Ultimately, this study lays a strong foundation for advancing the state-of the-art in intelligent and efficient web-based threat detection.

REFERENCES

[1] Salam, A.; Ullah, F.; Amin, F.; Abrar, M. Deep Learning Techniques for Web-Based Attack Detection in Industry 5.0: A Novel Approach. *Technologies***2023**, *11*, 107. https://doi.org/10.3390/technologies11040107

[2] Ullah, F.; Salam, A.; Abrar, M.; Ahmad, M.; Ullah, F.; Khan, A.; Alharbi, A.; Alosaimi, W. Machine health surveillance system by using deep learning sparse autoencoder. *Soft Comput.***2022**, *26*, 7737–7750

[3] Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; Part of Advances in Neural Information Processing Systems. Volume 30.

[4] García, S.; Luengo, J.; Herrera, F. *Data Preprocessing in Data Mining*; Springer: Berlin/Heidelberg, Germany, 2015.

[5] Popoola, S.I.; Adebisi, B.; Hammoudeh, M.; Gui, G.; Gacanin, H. Hybrid deep learning for botnet attack detection in the internet-of-things networks. *IEEE Internet Things J.***2020**, *8*, 4944–4956.

[6] Jeong, J.; Mihelcic, J.; Oliver, G.; Rudolph, C. Towards an improved understanding of human factors in cybersecurity. In Proceedings of the 2019 IEEE 5th International Conference on Collaboration and Internet Computing (CIC), Los Angeles, CA, USA, 12–14 December 2019; pp. 338–345.

[7] Hu, J., Zhao, W., and Cui, Y. (2020). A Survey on SQL Injection Attacks, Detection and Prevention. *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*, 483-488. https://doi.org/10.1145/3383972.3384028

[8] Ray, D., and Ligatti, J. (2012). Defining code-injection attacks. *Proceedings of the 39th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, 179-190. https://doi.org/10.1145/2103656.2103678

[9] Rahali, A., and Akhloufi, M. A. (2021). MalBERT: Using Transformers for Cybersecurity and Malicious Software Detection. arXiv. https://doi.org/10.48550/ARXIV.2103.03806

[10] Long, Z. et al. (2024). A Transformer-based network intrusion detection approach for cloud security. *Journal of Cloud Computing*, 13(1). https://doi.org/10.1186/s13677-023-00574-9

[11] Chen, Z. et al. (2019). Research on SQL Injection and Defense Technology. *Lecture Notes in Computer Science*, 191-201. https://doi.org/10.1007/978-3-030-24268-8_18

[12] Kheddar, H. (2025) *Transformers and large language models for efficient intrusion detection systems: A comprehensive survey*, *arXiv.org*. Available at: https://arxiv.org/abs/2408.07583 (Accessed: 29 April 2025).

[13] Ma, M., Han, L. and Zhou, C. (2024) *Research and application of artificial intelligence based webshell detection model: A literature review*, *arXiv.org*. Available at: https://arxiv.org/abs/2405.00066 (Accessed: 29 April 2025).

[14] Rahman, Md.M. *et al.* (2024) *Roberta-BiLSTM: A context-aware hybrid model for sentiment analysis*, *arXiv.org*. Available at: https://arxiv.org/abs/2406.00367 (Accessed: 29 April 2025).

[15] Ahmad, Z. et al. (2020). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1). https://doi.org/10.1002/ett.4150

[16] Demilie, W. B., and Deriba, F. G. (2022). Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques. *Journal of Big Data*, 9(1). https://doi.org/10.1186/s40537-022-00678-0

[17] Shahid, M. (2023). Machine Learning for Detection and Mitigation of Web Vulnerabilities and Web Attacks. arXiv. https://doi.org/10.48550/ARXIV.2304.14451

[18] J., A., and Kathrine, G. J. W. (2018). An Intrusion Detection System Using Correlation, Prioritization and Clustering Techniques to Mitigate False Alerts. *Advances in Intelligent Systems and Computing*, 257-268. https://doi.org/10.1007/978-981-10-7200-0_23

[19] Tasdemir, K. et al. (2023). Advancing SQL Injection Detection for High-Speed Data Centers: A Novel Approach Using Cascaded NLP. arXiv. https://doi.org/10.48550/ARXIV.2312.13041

[20] Shi, F. et al. (2020). Network attack detection and visual payload labeling technology based on Seq2Seq architecture with attention mechanism. *International Journal of Distributed Sensor Networks*, 16(4), 155014772091701. https://doi.org/10.1177/1550147720917019

[21] Gopali, S. et al. (2024). The Performance of Sequential Deep Learning Models in Detecting Phishing Websites Using Contextual Features of URLs. *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, 1064-1066. https://doi.org/10.1145/3605098.3636164

[22] Chen, K. and Jia, J. (2025) *Network evasion detection with Bi-LSTM Model*, *arXiv.org*. Available at: https://arxiv.org/abs/2502.10624 (Accessed: 29 April 2025).

[23] Uddin, M.A. and Sarker, I.H. (2024) *An explainable transformer-based model for phishing email detection: A large language model approach*, *arXiv.org*. Available at: https://arxiv.org/abs/2402.13871 (Accessed: 29 April 2025).

[24] Wang, L., Weng, Y., and Yu, W. (2025). Anesthesia depth prediction from drug infusion history using hybrid AI. *BMC Medical Informatics and Decision Making*, 25(1). https://doi.org/10.1186/s12911-02502986-w

[25] Chen, S., and Liao, H. (2022). BERT-Log: Anomaly Detection for System Logs Based on Pre-trained Language Model. *Applied Artificial Intelligence*, 36(1). https://doi.org/10.1080/08839514.2022.2145642

[26] Yang, Z. et al. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding. arXiv. https://doi.org/10.48550/ARXIV.1906.08237

[27] Ivanova, M., and Rozeva, A. (2021). Detection of XSS Attack and Defense of REST Web Service – Machine Learning Perspective. *2021 The 5th International Conference on Machine Learning and Soft Computing*, 22-28. https://doi.org/10.1145/3453800.3453805

[28] Lin, L., and Hsiao, S. (2022). Attack Tactic Identification by Transfer Learning of Language Model. arXiv. https://doi.org/10.48550/ARXIV.2209.00263

[29] Wan, Z. (2023). Text Classification: A Perspective of Deep Learning Methods. arXiv. https://doi.org/10.48550/ARXIV.2309.13761

[30] Yates, A., Nogueira, R., and Lin, J. (2021). Pretrained Transformers for Text Ranking: BERT and Beyond. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 1154-1156. https://doi.org/10.1145/3437963.3441667

[31] Liu, M., Li, K. and Chen, T. (2020) *Deepsqli: Deep semantic learning for testing SQL Injection*, *arXiv.org*. Available at: https://arxiv.org/abs/2005.11728 (Accessed: 29 April 2025).

[32] Zhuo, Z. et al. (2021). Long short-term memory on abstract syntax tree for SQL injection detection. *IET Software*, 15(2), 188-197. https://doi.org/10.1049/sfw2.12018

[33] Devlin, J. et al. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv. https://doi.org/10.48550/ARXIV.1810.04805

[34] Rahali, A., and Akhloufi, M. A. (2021). MalBERT: Using Transformers for Cybersecurity and Malicious Software Detection. arXiv. https://doi.org/10.48550/ARXIV.2103.03806

[35] Choudrie, J. et al. (2021). Applying and Understanding an Advanced, Novel Deep Learning Approach: A Covid 19, Text Based, Emotions Analysis Study. *Information Systems Frontiers*, 23(6), 1431-1465. https://doi.org/10.1007/s10796-021-10152-6

[36] Al-Garadi, M. A. et al. (2021). Text classification models for the automatic detection of nonmedical prescription medication use from social media. *BMC Medical Informatics and Decision Making*, 21(1). https://doi.org/10.1186/s12911-021-01394-0

[37] Xu, Z. (2021). RoBERTa-wwm-ext Fine-Tuning for Chinese Text Classification. arXiv. https://doi.org/10.48550/ARXIV.2103.00492

[38] Cui, Y., Yang, Z., and Liu, T. (2022). PERT: Pre-training BERT with Permuted Language Model. arXiv. https://doi.org/10.48550/ARXIV.2203.06906

[39] Sornsuwit, P., and Jaiyen, S. (2019). A New Hybrid Machine Learning for Cybersecurity Threat Detection Based on Adaptive Boosting. *Applied Artificial Intelligence*, 33(5), 462-482. https://doi.org/10.1080/08839514.2019.1582861

[40] He, Y., Deng, K., and Han, J. (2025). Patent value prediction in biomedical textiles: A method based on a fusion of machine learning models. *Plos One*, 20(4), e0322182. https://doi.org/10.1371/journal.pone.0322182

[41] Hu, J. *et al.* (2024) *Deep learning for medical text processing: Bert Model Fine-tuning and comparative study*, *arXiv.org*. Available at: https://arxiv.org/abs/2410.20792 (Accessed: 29 April 2025).

[42] Al-Garadi, M. A. et al. (2021). Text classification models for the automatic detection of nonmedical prescription medication use from social media. *BMC Medical Informatics and Decision Making*, 21(1). https://doi.org/10.1186/s12911-021-01394-0

[43] Phu, A. T. et al. (2023). Defending SDN against packet injection attacks using deep learning. arXiv. https://doi.org/10.48550/ARXIV.2301.08428

[44] Demilie, W. B., and Deriba, F. G. (2022). Detection and prevention of SQLI attacks and developing compressive framework using machine learning and hybrid techniques. *Journal of Big Data*, 9(1). https://doi.org/10.1186/s40537-022-00678-0

[45] Patterson, D. et al. (2021). Carbon Emissions and Large Neural Network Training. arXiv. https://doi.org/10.48550/ARXIV.2104.10350

[46] Long, Z. et al. (2024). A Transformer-based network intrusion detection approach for cloud security. *Journal of Cloud Computing*, 13(1). https://doi.org/10.1186/s13677-023-00574-9

[47] Patel, H., Rehman, U. and Iqbal, F. (2024) *Evaluating the efficacy of large language models in identifying phishing attempts*, *arXiv.org*. Available at: https://arxiv.org/abs/2404.15485 (Accessed: 29 April 2025).