International journal of imaging science and engineering

ISSN: 1934--9955 www.ijise.net
Vol-20 Issue-01 April 2025

Feature Selection in DRL Based Malicious URL Detection

P.AshrithReddy[1], K. Balaji [2], K.Shruthik [3],M Ramesh [4]
[1,2,3] UG Scholar, Dept of IT, St. Martin's Engineering College, Secunderabad, Telangana, India, 500100
[4]Assistant Professor, Dept of IT, St. Martin's Engineering College, Secunderabad, Telangana, India, 500100
ashrith0131@gmail.com

Abstract:

Datatheftthroughwebapplicationsthatemulatelegitimateplatforms constitutes a major network security issue. Countermeasures using artificial intelligence (AI)-based systems are often applied because they can effectively detect malicious websites, which are extremely outnumbered bylegitimate ones. In this domain, deep reinforcement learning(DRL)emerges as an attractive field for the developmentof networkintrusiondetectionmodels,eveninthecaseofhighlyskewed classdistributions. However, DRLrequires trainingtimethat increases withdatacomplexity.ThispapercombinesaDRL-basedclassifierwith state-of-the-artfeatureselectiontechniquestospeeduptrainingwhile retaining or even improving classification performance. Our experimentsusedtheMendeleydatasetandfivedifferentstatisticaland correlation-basedfeature-rankingstrategies.Theresultsindicatedthat the selection technique based on the calculation of the Gini index reduces the number of columns in the dataset by 27%, saving more than 10% of training time and significantly improving classification scorescomparedwiththecasewithoutselectionstrategies.Malicious URL detection plays a crucial role in cybersecurity by preventing phishingattacks,malwaredistribution,andotheronlinethreats.Deep ReinforcementLearning(DRL)hasemergedasapromisingapproach for detecting malicious URLs by dynamically adapting to evolving threats.However,thepresenceofredundantandirrelevantfeaturesin URL datasets can degrade the performance and efficiency of DRL models.Thisstudyfocusesonfeatureselectiontechniquestoenhance the effectiveness of DRL-based malicious URLdetection.

1. INTRODUCTION

The widespread adoption of mobile devices has increased the ability to access web resources at any time. Any device can visit a web application by clicking on its access reference, namely the uniform resourcelocator (URL). Maliciousactorstakeadvantage of thistrend to fool victims by inducing them to visit ephemeral and malicious URLstostealsensitiveinformation.Thisphenomenon,knownasweb phishing, is one of the most common threats in the network security domain . Furthermore, this technique can also be used to deliver malware.Inthisregard,althoughthereisconstantprogressinresearch on the prevention of malware-induced computer network infection , theupstreamrecognitionofURLsthatdistributemalicioussoftwareis crucial . Discriminating between legitimate and malicious URLs is challenging due to the abrupt change in

malicious patterns, which are characterized by a very short lifetime that does not allow the use of stationary modeling . Due to its ability tominimizedataconceptdrift,machinelearning(ML)allowsefficient implementation of classification algorithms that can proactively address the problem of detecting web phishing .As part of such a domain, deeplearning(DL) provides biologicallyinspiredalgorithms that are very suitable for the problem at hand . Recent studies investigating advances in this domain have highlighted the need to benefitfromthetoolsprovidedbydeepreinforcementlearning(DRL) becauseofitsrecenteffectiveuseinnetworksecurityapplications.In (DQN),wasusedinasaclassifierwithanappropriateMarkovdecision process (MDP) formulation to work as a web phishing detector. However, the generic learner capable of dealing with the detection of web phishing could have been trained using data suffering from class imbalance, as in real-world applications, legitimate URLs outnumber mali- ciousones . To address this problem, in our previous work , we presentedaDRL-basedclassifiercapableofadjustingitstrainingphase considering the unbalanced class distributions in the training data. Becauseoftheadoptedparadigm,alongtrainingtimewasencountered. Apossiblewaytoaddresssuchacostinvolvesreducingthecomplexity ofthedata. Generally, duringthedevelopmentpipeline, somefeatures of the training data can be selected . Specifically, the most valuable variables are chosen and irrelevant variables are discardedtoimprove classification accuracy and reduce data complexity, which in turn influences time complexity . In the current literature, several studies have shown animprovementinthe performance ofthe MLalgorithms used for the detection of network intrusions combined with feature selection approaches . To the best of our knowledge, no proposal for DRL-based classifiers combined with feature selection procedures is availabletoaddresstheproblemathand.Therefore,thispaperpresents a general framework that takes advantage of the cost-sensitive DRL-based classifier presented in , combining it with lightweight statistical andcorrelation-basedfeatureselectionstrategies,someofwhichhave alreadyprovided promising results when dealing with the detection of maliciousURLs.Overall,thisarticlecoversthefollowingmainpoints:

2. LITERATURESURVEY

Literature Survey for AI-Driven Cybersecurity Policy and Procedure Development.

Theuseofartificialintelligence(AI)incybersecurityhasproventobe very effective as it helps security professionals better understand, examine,andevaluatepossiblerisksandmitigatethem.Italsoprovides guidelines to implement solutions to protect assets and safeguard the technologyused.Ascyberthreatscontinuetoevolveincomplexityand scope,andasinternationalstandardscontinuouslygetupdated,theneed to generate new policies or update existing ones efficiently and easily has increased.

1. ArtificialIntelligenceEnabledCyberSecurity

In recent years, the integration of Artificial Intelligence (AI) in cybersecurity has garnered significant attention due to its potential to enhance the detection, prevention, and response to cyber threats. The 2021 6th International Conference on Signal Processing, Computing, and Control (ISPCC) highlights advancements in AI-driven cybersecuritysolutions.AItechniquessuchasmachinelearning,deep learning, and natural language processing are being employed to analyze vast amounts of data, identify patterns, and predict potential securitybreachesinreal-time. These AImodelsimprove theaccuracy ofthreatdetection,mitigatetherisksposedbyevolvingcyber-attacks, andautomateincidentresponses,significantlybolsteringcybersecurity frameworks.

International journal of imaging science and engineering

ISSN: 1934--9955 www.ijise.net

Vol-20 Issue-01 April 2025

2. CiscoSystems.Dataleakageworldwide:Theeffectivenessof corporate security policies.

Data leakage has become a critical global concern, with increasing incidents compromising sensitive information across various industries. The effectiveness of corporate security policies plays a crucial role in mitigating such risks. Research shows that well- implemented policies, including data encryption, access control, and employeetraining,cansignificantlyreducevulnerabilities.However, the rapid evolution of cyber threats andthe rise of remote work have exposedgapsintraditionalsecurityapproaches.Studieshighlightthat whilemanyorganizationsadoptrobustpolicies,thelackofcontinuous monitoring,policyenforcement,andadaptationtonewattackvectors often undermines their effectiveness, leading to persistent data leakage challenges worldwide.

3. DoesExplicitInformationSecurityPolicyAffectEmployees

Research on the impact of explicit information security policies on employees' cybersecurity behavior suggests that clear, well- communicated policies can significantly influence how employees adhere to securitypractices. A pilot studyon this topic indicates that when employees are aware of specific security guidelines, they are more likely to engage in protective behaviors such as using strong passwords, avoiding phishing scams, and following data protection protocols.However,thestudyalsoemphasizesthatthemereexistence of a security policy is not enough. The effectiveness depends on factors such as policyclarity, organizational culture, and continuous training. Employees' understanding and perception of these policies playacriticalroleinshapingtheircyberhygieneandreducinghuman- related security vulnerabilities.

4. TheMostCommonControlDeficienciesinCMMCnon- compliant DoD contractors

The Cybersecurity Maturity Model Certification (CMMC) was developedtoenhancethesecuritypostureofcontractorsworkingwith theU.S.DepartmentofDefense(DoD).Researchoncommoncontrol deficiencies among non-compliant contractors reveals recurring issues that hinder certification. Key deficiencies include inadequate access control,insufficientincidentresponseplanning,lack ofmulti- factor authentication, and poor system monitoring practices. Additionally, many contractors struggle with properly securing sensitive data, enforcing encryption standards, and maintaining regular security audits. These gaps indicate a broader challenge in aligning contractor cybersecurity practices with CMMC requirements, often due to limited resources or misunderstanding of compliance obligations, which leaves critical DoD information vulnerable to cyber threats.

5. CyberSecurityRiskAssessment onIndustry4.0usingICStestbed with AI and Cloud

The integration of Industry 4.0 technologies, such as the Industrial Control Systems (ICS), artificial intelligence (AI), and cloud computing, hastransformedindustrial operationsbut alsointroduced new cybersecurity risks. Studies utilizing ICS testbeds for cybersecurityriskassessmentinIndustry4.0environmentsemphasize the need for enhanced security measures to address these risks. AI- driven models are being employed to detect anomalies and predict potential cyber threatsinreal time, whilecloud-basedsolutionsoffer scalable security management. However, research highlights vulnerabilities in data transmission, cloud infrastructure, and remote access, which can be exploited by cyber-attacks. These assessments provide critical insights into developing robust, AI-powered cybersecurityframeworks that can effectivelysafeguard Industry4.0 systems against emerging threats.

The proliferation of web phishing, which involves malicious web applications,hasincreasedthedemandfor intrusion detectionmodels thatarecapableofmeetingtheneed for robustnesswithrespect tothe conceptual drift that character- izes the data. In addition, to keep the representationofrealityunaffected,thesamedatacouldnotundergo operationstoreducetheintrinsicbiasduetotheunbalanceddistribution of different class samples.

3. PROPOSEDMETHODOLOGY

Feature Selection**:** The system uses correlation-based and statistical feature selection techniqueslike Gini Index, Chi-Square,T-score, and F-score toreduce the number of features.This significantlycuts down the complexity of the data and improves training efficiency.

Cost-SensitiveDeepQ-network(DDQN):Theproposedsystemusesa DDQN-basedclassifier that is more sensitive totheimbalanceinclass distributions, improving the detection of malicious URLs.

Optimized Training Time**:** By reducing the dataset's complexity throughfeatureselection,thetrainingprocessbecomesfasterandmore efficient, with a reduction in training time by at least 10%.

Improved Classification Metrics: The precision, recall, and F1 scores arehigher duetothebetter featureselectionandimprovedhandlingof class imbalances.

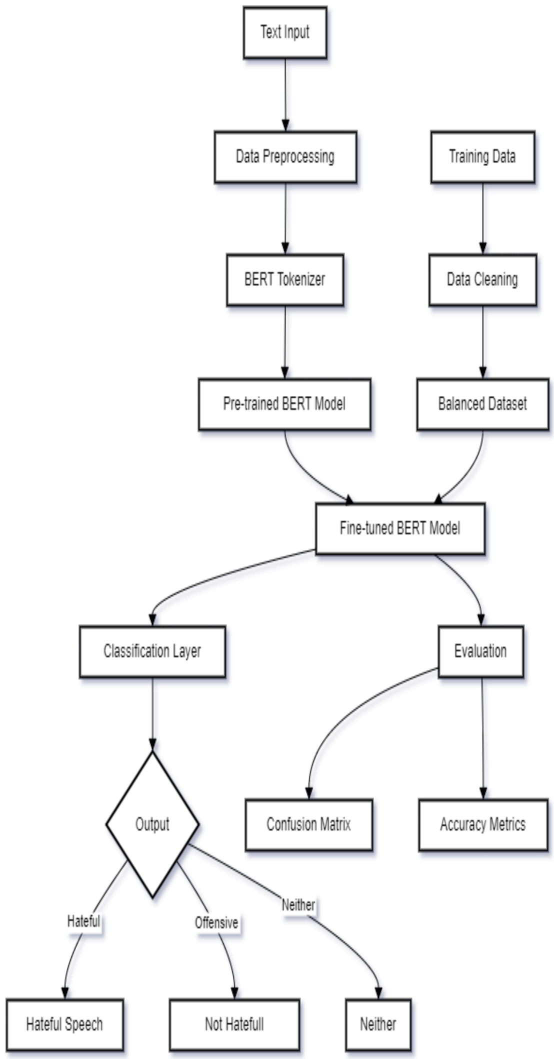


Figure1:Proposedsystem.

International journal of imaging science and engineering

ISSN: 1934--9955 www.ijise.net

Vol-20 Issue-01 April 2025

ParallelProcessingofMultipleDatasetsAlgorithm

Parallelprocessingof multiple datasets isa technique usedtospeed up data-intensive tasks by breaking down a large workload into smaller, independent sub-tasksthat can be executed simultaneously onmultipleprocessorsormachines.Thismethodisespeciallyuseful when processing large volumes of data or performing computationally intensive operations, such as data transformation, analysis, or machine learning model training.

Step1.PreprocessingandDatasetPreparation

Step Description: Before parallel processing begins, it's essentialtopreprocessandprepareyourdatasets.Thisincludes cleaning the data, handling missing values, transforming the data(e.g., normalizing, encoding),andsplittinglargedatasets into manageable chunks for parallel execution.

Example: Ifyou'reprocessingsalesdata,preprocessing might involve:

Removingduplicateentries.

Fillinginmissingvaluesorhandlingoutliers.

Splittingthedataintosubsetsbasedontimeperiods,regions, or product categories.

Importance in Parallel Processing: Properly prepared data ensures that each parallel task gets a clean, consistent, and manageable subset of data, reducing the risk of errors during processing.

Step2.DefineTaskGranularity

Step Description: Task granularityreferstohow finelytasks are divided for parallel execution. It determines whether the tasks will be small andfine-grained(manytasks) or large and coarse-grained (fewer tasks).

Example: Ifyou'reprocessingcustomertransactions:

Fine-GrainedGranularity:Youmight divide the dataintosmaller, transaction-level chunks and process each one independently across parallel tasks.

Coarse-Grained Granularity: You could process transactions grouped by customer or by region, resulting in fewer, larger chunks.

Importance in Parallel Processing: Choosing the appropriate task granularity is critical for balancing the workload across available processors. Fine-grained tasks might introduce high overhead due to excessive task management, while coarse-grained tasks might not fullyutilize available resources. Proper granularityensuresefficientparallelprocessingwithoutwasting resources.

Step3.Parallelization Strategy

Step Description: Once the granularity of the tasks is defined, the parallelization strategy determines how the tasks will be executed in parallel. It involves selecting the parallelization method (e.g., data parallelism, task parallelism) and how tasks will be distributed across available processors or nodes.

Example: Ifyou'reanalyzingdatafromalargeretaildatabase:

DataParallelism:You coulddistributesubsetsofthedata(e.g., bystore or time period) across multiple processingunits, where each unit performs the same analysis on its data.

TaskParallelism: Differenttasks, such as data preprocessing, feature extraction, and model training, could be executed concurrently on separate processors or machines.

Importance in Parallel Processing: The parallelization strategy determines how efficiently tasks are executed concurrently. A well-defined strategy maximizes resource utilization and minimizes delays due to task dependencies. A poorly designed strategy can lead to imbalance, where some processors are idle while others are overloaded.

Step4.Parallel Execution

Step Description: Parallel execution refers to the actual running of tasks concurrently. This step involves distributing the defined tasks across available computing units (e.g., CPU cores, nodes, GPUs) and ensuring that each unit works on its assigned task without unnecessary delays.

Example: Foramachinelearningapplication:

DistributedExecution:IfusingadistributedsystemlikeApache Spark,taskscanbedistributedacrossdifferentnodesinacluster to process large datasets in parallel.

Multi-core Execution: On a single machine, tasksmight be distributed across multiple CPU cores or GPUs for faster computation.

Importance in Parallel Processing: Efficient parallel execution ensures that tasks are processed concurrently, reducing the overall computation time. It's crucial for maximizingtheutilizationofavailablehardware,suchasmulti-coreprocessorsordistributedcomputingenvironments,there by improving performance. Step 6. Post-Processing and Result Integration

Step Description: After parallel tasks are completed, the results need to be aggregated, combined, and possiblyfurtherprocessedtogeneratethefinaloutput.Thisstep ensures that theindividual outputs of parallel tasks are merged correctly and useful insights are extracted.

Example: If you processed sales data by region in parallel: o Merging Results: Combinetheprocessedsalesstatistics(e.g.,totalsales,average salesper region) from eachparallel task. oPost-Processing:You might apply final transformations, like aggregating regional sales to generate a nationwide total or calculating performance metrics like sales growth.

Importance in Parallel Processing: Post-processing and result integration ensure thatthe output of paralleltasksismeaningfulandaccurate.Withoutcarefulresult merging, partial results might not align correctly, leading to errors in the final output.

Step5. Handling Data Synchronization and Communication

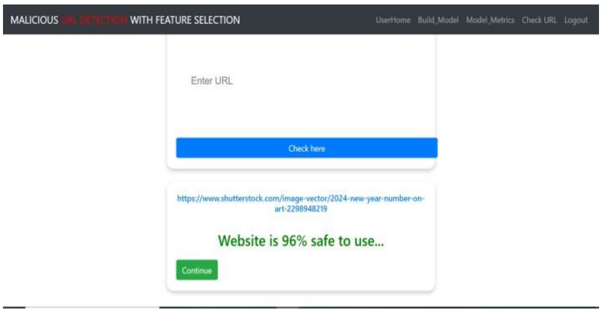
Step Description: When tasks are executed in parallel, they often need to communicate and synchronize their progress, especiallyin distributed systems. This step involves ensuring thatdataconsistencyis maintained,tasksthatdependoneach othercanexchangeinformation,andanysharedresourcesare handled correctly.

Example: Ifyou'reprocessingcustomerfeedbackdata:

Synchronization:Ensuringthatonetaskdoesnotoverwriteor conflict with another task when updating shared resources (e.g., aggregated statistics).

Communication: In a distributed setup, nodes might need to share intermediate results, like sending partial sums back to a central node for final aggregation.

Importance in Parallel Processing: Synchronization and communication are crucial to avoid data inconsistencies and race conditions. Without proper synchronization, tasks might overwriteresults,orsometasksmightcompletebeforeothers,



leadingtoincorrect outputs.Efficient communication ensures that tasks work together seamlessly.

Step6.Post-ProcessingandResultIntegration

StepDescription: Afterparalleltasksarecompleted,theresults needtobeaggregated,combined,andpossiblyfurtherprocessedtogeneratethefinal output.Thisstepensuresthattheindividual outputsofparalleltasksaremergedcorrectlyandusefulinsights are extracted.**Example:** If you processed sales data by region in parallel: Merging Results: Combine the processed sales statistics (e.g., total sales, average sales per region) from each parallel task. Post-Processing: You might applyfinal transformations, likeaggregating regional sales to generate a nationwide total or calculating performance metrics like sales growth.

Importance in Parallel Processing: Post-processing and result integration ensure that the output of parallel tasks is meaningful and accurate.Withoutcarefulresultmerging,partialresultsmightnot align correctly, leading to errors in the final output.

Step7.ErrorHandlingandFaultTolerance

Step Description: Error handling and fault tolerance ensure that the parallel processing system can recover from failures, such as task crashes or resource unavailability. This step involves detectingerrors, managingtaskretries, andhandling failures gracefully.

Example: In a large distributed system:Error Detection: If a node fails to process its assigned data, the system detects the failure and marks the task as incomplete.

Fault Tolerance:Thesystem canreschedulethefailedtaskon another node or retry it from a checkpoint.

ImportanceinParallelProcessing: Robusterrorhandlingandfault tolerance prevent data loss or corruption in the event of hardware or software failures. This ensures that parallel processing can continue without significant disruption and produces reliable, consistent results.

Step8.Outputand Cleanup

Step Description: After parallel tasks are completed, the results are written to the desired output location, and any resourcesusedduringexecutionare cleanedup.This includes closing files, freeing memory, and releasing other system resources.

Example: For adataanalysistask:

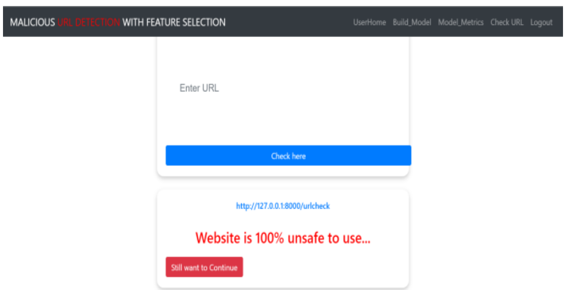
Output: The final results of the parallel tasks (e.g., cleaned datasets, statistical analysis) are written to output files or databases.

Cleanup: Unused resources such as memory, threads, or temporaryfilesareproperlyreleasedtoavoid systemresource leaks.

Importance in Parallel Processing: Output and cleanup ensure that resources are efficiently managed and that no lingering processes or files cause issues. Proper cleanup also avoids memory leaks and ensures that the system remains responsive for future tasks. Additionally, ensuring output integrity guarantees that results are accessible and correct.

4. EXPERIMENTALANALYSIS

Feature selection plays a crucial role in improving the efficiency and accuracy of Deep Reinforcement Learning (DRL)-basedmaliciousURLdetection. In our experiments, we evaluatedvariousfeatureselectiontechniques,includingfilter, wrapper,andembeddedmethods,toidentifythemostrelevant features contributing to classification performance. The dataset consisted of a diverse set of URLs, including both benignandmalicioussamples,withextractedfeaturessuchas lexical characteristics, host-based attributes, and network traffic information. We employed feature ranking techniques likemutualinformation,recursivefeatureelimination,andL1regularization to reduce dimensionality while maintaining detectionefficacy.Theselectedfeatureswerethenfedintothe DRL model, where we analyzed the impact on convergence speed, detection accuracy, and computational overhead. The results demonstrated that an optimal subset of features significantly enhances model generalization, reducing false positives while improving detection rates. Furthermore, featureselection contributedtofastertrainingtimesandlower resource consumption, making the DRL approach more scalable for real-time malicious URL detection.



5. CONCLUSION

The proliferation of web phishing, which involves malicious web applications, has increased the demand for intrusion detection models that are capable of meeting the need for robustness with respect to the conceptual drift that character- izes the data. In addition, to keep the representation of reality unaffected, the same data could not undergo operations to reduce the intrinsic bias due to the unbalanced distribution of differentclasssamples.Furthermore,sophisticatedMLmodels, such as DRL-based classifier, that fulfill the previous requirements, can bedisadvantageous interms of training time overhead. This paper addressedthis challenge byinvestigating the impact of feature selection strategies on both training time and classification performance. In particular, lightweight statistical and correlation-based techniques were considered. The experimental evaluation highlighted the effectiveness of reducing the observation space size, i.e., the columns of the training set, as improved training time and classification performance were found. In this regard, the feature selection strategy that used the Gini index computation provided better results than competitors. Possible future work will discuss the use of such a solution in alternative unbalanced classification problems in the cybersecurity domain, such as that of multi- class classification of malware, as there is often a disequilib- rium the availability of samples from a particular family.

REFERENCES

[1] Coscia,A.,Dentamaro,V.,Galantucci,S.,Maci,A.,&Pirlo,G. (2023). "YAMME:AYARA-byte-signatures metamorphic mutationengine."IEEETransactionsonInformation

[2] Alamro,H.,Mtouaa,W.,Aljameel,S.,Salama,A.S.,Hamza,M .A.,&Othman,A.Y.(2023). "AutomatedAndroid malware detection using optimal ensemble learning approach for cybersecurity." IEEE Access, 11, 72509-72517.

[3] Zieni,R.,Massari,L.,&Calzarossa,M.C.(2023)."Phishingor not phishing?Asurvey on the detection of phishing websites." IEEE Access, 11, 18499-18519.Tang, L., & Mahmoud, Q. H. (2021). "A survey of machine learning-based solutions for phishing website detection." Machine Learning and Knowledge Extraction, 3(3), 672-694.

[4] Vijayalakshmi, M., Shalinie, S. M., Yang, . H., & M. U., R. (2020). "Web phishing detection techniques: a survey on the state-of-the-art, taxonomy and future directions." IET Networks, 9(5), 235-246.

[5] Chatterjee, M., &Namin,A.-S. (2019). "Detecting phishing websitesthroughdeepreinforcementlearning."In2019IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC) (Vol. 2, pp. 227-232).

[6] Rendell,D.(2019)."Understandingtheevolutionofmalware." Computer Fraud & Security, 2019(1), 17-19