PHISHING URL DETECTION: A REAL-CASE SCENARIO THROUGH LOGIN URLS

V.Savithri[1],V.Surendra[2]

[1]Assistant Professor, Dept. of MCA, Audisankara College of Engineering & Technology (AUTONOMOUS), Gudur, AP, India.

[2]PG Scholar, Dept. of MCA, Audisankara College of Engineering & Technology (AUTONOMOUS),Gudur, AP, India.

ABSTRACT

Phishing is a social engineering cyberattack where criminals deceive users to obtain their credentials through a login form that submits the data to a malicious server. In this paper, we compare machine learning and deep learning techniques to present a method capable of detecting phishing websites through URL analysis. In most current state-of-the-art solutions dealing with phishing detection, the legitimate class is made up of homepages without including login forms. On the contrary, we use URLs from the login page in both classes because we consider it is much more representative of a real case scenario and we demonstrate that existing techniques obtain a high false-positive rate when tested with URLs from legitimate login pages. Additionally, we use datasets from different years to show how models decrease their accuracy over time by training a base model with old datasets and testing it with recent URLs. Also, we perform a frequency analysis over current phishing domains to identify different techniques carried out by phishers in their campaigns. To prove these statements, we have created a new dataset named Phishing Index Login URL (PILU-90K), which is composed of 60K legitimate URLs, including index and login websites, and 30K phishing URLs. Finally, we present a Logistic Regression model which, combined with Term Frequency - Inverse Document Frequency (TF-IDF) feature extraction, obtains 96.50% accuracy on the introduced login URL dataset.

1. INTRODUCTION

In the last years, web services usage has grown drastically due to the current digital transformation. Companies motivate the change by providing their services online, like e-banking, e-commerce or SaaS (Software as a Service) [1]. Nowadays, due to the COVID-19 pandemic, restrictions have spread out the work-from-home model, which implies extra millions of workers, students, and teachers developing their activities remotely [2], leading to a substantial additional workload for services such as email, student platforms, VPNs or company portals. Therefore, there are even more potential targets exposed to phishing attacks, where phishers try to mimic legitimate websites to steal users' credentials or payment information [3], [4]. Recent studies [5], [6] concluded that phishing is one of the most significant attacks based on social

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar .engineering during the COVID-19 pandemic, together with spam emails and websites to execute these attacks.

Identifying phishing sites through their HTTP protocol is no longer a valid rule. In the 3rd quarter of 2017 [7], the APWG reported that less than 25% of phishing websites were hosted under HTTPS protocol, whilst this amount has increased up to 83% in 1st quarter of 2021 [8]. These websites provide secure end-to-end communication, which transmits a false safe impression to the user while making an online transaction [9]. Furthermore, the Anti- Phishing Working Group (APWG) [10] has reported a significant increase in phishing attacks, i.e. from 165, 772 to 611, 877 websites, just between the first quarter of 2020 and 2021 respectively. A reason behind this increase might be that people have resorted (and still are) to online services during the COVID-19 pandemic.

One of the most popular solutions for phishing detection is the list-based approach, which analyzes the requested URL against a phishing database [11]. Some examples of this solution are Google SafeBrowsing,1 PhishTank,2 OpenPhish3 or SmartScreen.4 If a requested URL matches any record, the request is blocked, and a warning is displayed to the user before visiting the website. However, despite the capabilities of the list-based approach, it would fail if the phishing URL was not reported previously [12]–[14], and it will require a continuous effort to update the database with newer phishing data. Bell and Komisarczuk [11] observed that many phishing URLs were removed after day five from Phishtank while OpenPhish removed all URLs after seven days from its report. This issue allows attackers to reuse the same URL when it is removed from different lists.

Due to the mentioned drawbacks with the blacklist-based methods, automatic detection of phishing URLs based on machine learning, have attracted attention in research [15], [16]. These approaches can be grouped into four classes according to the type of data used for the detection: the text of the URL, the page content, the visual features and networking information [17]. Methods based on the page content and visual features require visiting the website to collect the source code and render it, which is a time-consuming task. Other availability limitations can be found in studies that rely on networking and 3rd party information such as WHOIS or search engine rankings. To overcome these limitations, we focus on phishing detection through URLs since it implies advantages such as fast computation -because no websites are loaded- and 3rd party and language independent, since features are extracted only from the URLs.

2. LITERATURE SURVEY

In the literature, researchers have focused on phishing detection following three main approaches: List-based and automatic detection using Machine Learning and Deep Learning techniques.

A. LIST-BASED

The list-based approach, well-known for detecting phishing URLs [22]–[24], can be based on whitelists or blacklists, depending if they store legitimate or phishing URLs, respectively. Jain and Gupta [24] developed a whitelist-based system that blocks all websites which are not on that list. Conversely, the blacklist-based systems, like Google Safe Browse or PhishNet [23], are more common as they provide a zero false-positive rate, i.e. no legitimate website is classified as phishing. However, they can be compromised if an attacker makes changes on a blacklisted URL. Besides, they depend heavily on the update rate of the system's records. Therefore, a list-based approach is not a robust solution due to the high volume of new phishing websites introduced daily and their short lifespan, which is estimated to be 21 days on average [12].

B. MACHINE LEARNING METHODS

To overcome blacklist disadvantages, researchers have devel- oped machine learning models to detect unreported phishingencounters. Depending on their input data, these approaches can be classified into two categories: URL-based and content- based.

1) URL-BASED

Buber et al. [25] implemented a URL detection system com- posed of two sets of features. The first was a 209 word vector, obtained with "StringToWordVector" tool from Weka.6 The second, 17 NLP (Natural Language Processing) handcrafted features such as the number of sub-domains, random words, digits, special characters and length measurements over the URL words. Combining both feature sets, they obtained a high 97.20% accuracy with Weka's RFC (Random Forest Classifier) on a 10% sub-sample set from Ebbu2017 dataset. In the following studies, Sahingoz et al. [21] defined three different feature sets: Word vectors, NLP and a hybrid set combining both sets. They obtained a 97.98% accuracy on Random Forest (RF) using only 38 NLP features on Ebbu2017 [25] dataset. In this work, we used the NLP features from Sahingoz et al. [21], since they reported state- of-the-art performance in the last studies.

Jain and Gupta [26] built an anti-phishing system using 14 handcrafted URL descriptors, including some obtained using 3rd party services like WHOIS registers or DNS lookups. They obtained an accuracy of 76.87% and 91.28% with Naìve Bayes (NB) and Support Vector Machine (SVM) classifiers, respectively, on a private dataset with 35, 491 samples.

Banik and Sarma [27] implemented a lexical feature selection from URL to optimize the number of features and the accuracy of their model. They started with a set of 17 descriptors and removed the less significant ones until they reached an optimal performance. Using 9 features and a Random Forest (RF) classifier they obtained 98.57% accuracy on an extension of PWD2016 [18] dataset.

2) CONTENT-BASED

Content-based works use features extracted mainly from the websites' source code. However, most of the current works combine these with URLs and other 3rd party services such as WHOIS [28], [29].

One of the first content-based works was CANTINA [30], which consists of a heuristic system based on TF-IDF. CANTINA extracts five words from each website using TF- IDF and introduced them into the Google search engine. If a domain was within the n first results, the page was considered legitimate, or phishing otherwise. They obtained an accuracy of 95% with a threshold of n 30 Google search results. Due to the use of external services like WHOIS7 and the high false-positive rate, authors proposed CANTINA [31]. Their new proposal achieved a 99.61% F1-Score including two filters: (i) a comparison of hashed HTML tags with known phishing structures and (ii) the discarded websites with no form.

Moghimi and Vorjani [32] proposed a system independent from third services like Google Page Rank or WHOIS. They used two handcrafted feature sets, extracted from the URL and the Document Object Model (DOM) of the website. The first set has nine legacy features including a set of keywords, while the second has eight novel features which inform of whether the website's resources are loaded using SSL protocol or not. They used Levenshtein distance [33] to detect typo-squatting by comparing the website and resources URLs. These features were used to train an SVM classifier and obtained an accuracy of 98.65% on their banking websites dataset.

Adebowale et al. [34] created a browser extension to protect users by extracting features from the URL, the source code, the images, and features extracted using third- party services like WHOIS. Those features were introduced into an Adaptive Neuro-Fuzzy Inference System (ANFIS) and combined with the Scale-Invariant Feature Transform (SIFT) algorithm, obtaining an accuracy of 98.30% on Rami et al. [35] dataset.

Rao and Pais [28] developed a phishing website classifier using the URL, the hyperlinks on the HTML code and third-party services including the age of the domain and the page rank on Alexa. They reached 99.31% accuracy with a Random Forest classifier.

Yang et al. [36] proposed an Extreme Learning Machine (ELM) model and established three different groups of features: (i) Surface features, composed of 12 URL handcrafted and 4 Domain Name System (DNS) features related to the registration date and the DNS records for that domain; (ii) 28 Topological features that are related to the structure of the website and (iii) 12 deep features related to the text and image similarity. Combining these sets of features and the ELM classifier, they obtained 97.5% accuracy.

Sadique et al. [37] presented a framework for real-time phishing detection using four sets of URL features: (i) Lexical features related to the number of characters, dots and symbols found in different parts of the URL, (ii) host-based features that are related to the host, (iii) WHOIS features are related to the registration date and (iv) GeoIP-based features like the Autonomous System Number (ASN). A total of 142 individual features were evaluated using 98, 000 samples from Phishtank, where legitimate samples are also picked from false positives collected at PhishTank. They obtained a 90.51% accuracy on a Random Forest classifier using the proposed descriptors.

Li et al. [29] presented a stacking model which was the combination of three models: Gradient Boost Decision Tree (GBDT), eXtreme Gradient Boosting (XGBoost) and Light Gradient Boosting Model (LGBM). This stacking model was fed with a set of features from different sources: eight from the URL, 11 from the HTML and HTML string embeddings inspired by Word2Vec model [38]. They obtained 97.30% accuracy using a 49, 947 samples dataset.

C. DEEP LEARNING

Regarding the methods based on Deep learning, Some- sha et al. [39] proposed a model based on Long Short- Term Memory (LSTM) to classify phishing URLs using ten handcrafted features from Rao and Pais [28]. Those features are three URL features based on the number of dots, the length of the URL, and the presence of HTTPS, six features extracted from the HTML, including the internal links and images, the ratio of broken links and the presence of anchor links on the HTML body. Finally, one third-party numeric feature was obtained from Alexa’s Page Rank. These features were extracted from a 3, 526 samples dataset and introduced into the LSTM model to obtain 99.57% accuracy.

Aljofey et al. [40] presented an RCNN model to classify phishing URLs. They used the URL as input for a tokenizer and then used a one-hot encoding to represent the URL as a matrix at a character level. The last step is to set a fixed length of 200 characters for the model input. If the URL is under that threshold, the remaining characters are filled with zeros. Otherwise, the characters above the limit are trimmed. Finally, they used a 310, 642 URL dataset to feed an RCNN model, which obtained 95.02% using the aforementioned character embedding level features.

Al-Alyan and Al-Ahmadi [41] proposed a modified Convolutional Neural Network (CNN). First, they omitted the URL protocol and then cropped URLs larger than 256 characters. They used a 69 characters alphabet with lower-case letters, numbers and some symbols to obtain a 128 embedding vector. Then, a one-dimensional CNN was applied to obtain 95.78% accuracy on a 2, 307, 800 URLs dataset.

Zhao et al. [42] presented a Gated Recurrent Neural Net- work (GRU) capable of learning sequences and patterns within the URLs. They compared this approach against a set of 21 handcrafted features combined with an RF classifier. Results showed how automatic feature extraction combined with GRUs outperformed RF, reaching 98.5% and 96.4% respectively.

3. PROPOSED SYSTEM

The architecture of system for detecting phishing URLs is displayed in Figure. This system's main purpose is to determine whether a URL entered as input is a ligitimate URL or not. The two steps of the suggested methodology are (1) the URL search phase and (2) the feature extraction phase. When a user requests or accesses a URL during the URL search phase, a search is run to see if the requested URL is included in the repository of valid URLs. The URL is genuine if there is a match in the repository. If not, the URL is considered invalid and moves on to the next stage. The primary benefit of performing the search phase prior to the feature extraction phase is that it decreases the amount of computing that is required during the feature extraction phase and speeds up the system's overall reaction time. We have developed models to extract features from URLs during the feature extraction phase, and these models are then put through association rule mining to distinguish between authentic and phished URLs.

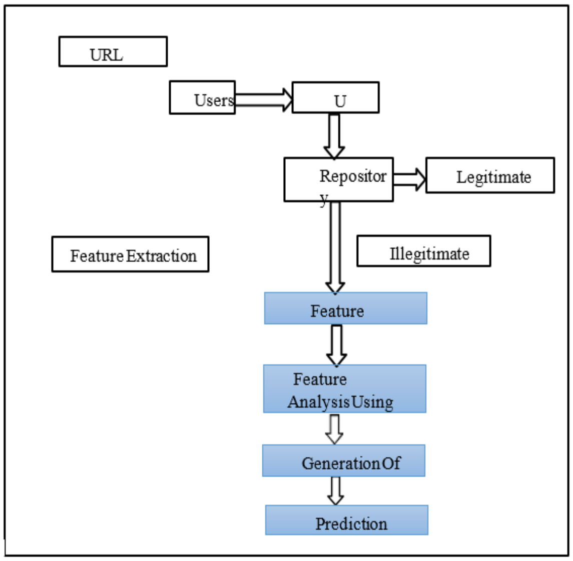

Fig.: System Architecture

MACHINE LEARNING ALGORITHMS TO DETECT PHISHING URL'S

Phishing URL detection can be done using a various machine learning algorithms and techniques. Several of the most common algorithms are listed below:

1. Logistic Regression
2. k-Nearest Neighbors
3. Support Vector Classifier
4. Naive Bayes
5. Decision Tree
6. Random Forest
7. Gradient Boosting
8. XGboost
9. Multilayer Perceptron

1. Logistic Regression: Logistic regression is one of the statistical method that is being utilised to model the probability of a binary outcome (i.e., a yes or no response) resulting from one or more predictor variables. It is a class of regression analysis where the dependent variable is categorical, and the independent variable(s) can be either categorical or continuous. In logistic regression, the outcome variable is model as aim of the predictor variables using a logistic function.

2.K-Nearest Neighbors Classifier: K-Nearest Neighbors is kind of the most basic classification algorithms in Machine Learning. It comes under the supervised learning technique and it is mostly used in both regression & classification problem.

3. Support Vector Machine (SVM): SVM is a popular machine learning algorithm. SVM works by finding the optimal hyperplane that separates data which divided into classes. in a way that maximizes the margin between the two classes. The hyperplane is chosen based on the support vectors, which are the data points closest to the decision boundary. SVMs probably used for binary classification as well as multiclass classification problems. In addition, SVMs can handle both linear and nonlinear data by using different typesof kernel functions.

4. Naive Bayes Classifier: Naive Bayes is a statistical method for machine learning that is commonly used for classification tasks. It is based

on Bayes' theorem, which is a fundamental principle in probability theory. The basic principle of Naive Bayes is to calculate the possibility of a data point involving a certain class given its features. This is done by estimating the probability distribution for each category , every characteristic using a training dataset, and then using these distributions to calculate the possibility of a new data point belonging to each class.

5. Decision Trees Classifier: The machine learning technique for classification and regression applications is the decision tree.. The decision tree algorithm starts by selecting the feature that provides the best split of the information based on the two categories in a certain criterion, such as maximizing information gain or minimizing entropy. This process is iterated again for each outcome group until a stopping criterion is met, such as a maximum tree depth or a particular quantity of samples per leaf.

6. Gradient Boosting Classifier: For classification issues, a form of machine learning method called a gradient boosting classifier is utilized It acts by generating a set of decision trees that are trained together of decision trees which has been trained using sequential manner, with each tree attempting to make the previous one's mistakes correct. The algorithm begins by creating a single decision trees and then making predictions based on that tree. The errors or residuals from the predictions are then used to trained a new decision tree, which is added to the ensemble. This process is repeated for a specified no. of repetitions or until a certain level of accuracy is achieved.

7. Random Forest: In machine learning, the commonly used ensemble learning methods for random forest is utilized for classification and regression applications. It involves creating multiple decision tree on randomly selected subsets of the trained data, and then aggregating the results of these trees to make a final prediction. Each decision tree in a random forest is constructed using a random subset of the data's features, which helps to reduce overfitting and improve the generalization performance of the model. Additionally, the random sampling of training data helps in minimizing variance and enhancing the model's overall accuracy.

8. XGBoost: A popular and advantageous machine learning strategy for regression and classification problems is called XGBoost (eXtreme Gradient Boosting). It is an implementation of gradient boosting, which involves building an ensemble of weak prediction models (usually decision trees) and iteratively improving them by minimizing a loss function. XGBoost has become popular because of its high prediction accuracy and efficiency. Additionally its ability to handle large-scale datasets. It

involves a number of techniques to optimize model performance, including regularization to prevent overfitting and parallel processing to speed up training.

9. Multi-layer Perceptron classifier: Multilayer Perceptron (MLP) is a form of neural network that consist of multiple layers of interconnected nodes or artificial neurons. In the MLP, every node take an input from nodes in the layer below and creates an output that is delivered to nodes in the layer previously. MLPs are a form of feedforward neural network, which implies that data moves from the input layer to output layer in a single direction.

4. CONCLUSION

Phishing detection mechanism aims to improve current blacklist methods, protecting users from malicious login forms. Our work provides an updated dataset PILU-90K for researchers to train and test their approaches. This dataset includes legitimate login URLs which are the most representative scenario for real-world phishing detection.

We explored several URL-based detection models using deep learning and machine learning solutions trained with phishing and legitimate home URLs. The main advantage of our approach is the low false-positive rate when classifying this type of URL.

5. REERENCES

[1] Statista. (2020). Adoption Rate of Emerging Technologies in Organizations Worldwide as of 2020. Accessed: Sep. 12, 2021. [Online]. Available: https://www.statista.com/statistics/661164/worldwide-cio-survey- operati%onal-priorities/

[2] R. De', N. Pandey, and A. Pal, ‘‘Impact of digital surge during COVID- 19 pandemic: A viewpoint on research and practice,'' Int. J. Inf. Manage., vol. 55, Dec. 2020, Art. no. 102171.

[3] P. Patel, D. M. Sarno, J. E. Lewis, M. Shoss, M. B. Neider, and C. J. Bohil, ‘‘Perceptual representation of spam and phishing emails,'' Appl. Cognit. Psychol., vol. 33, no. 6, pp. 1296–1304, Nov. 2019.

[4] J. A. Chaudhry, S. A. Chaudhry, and R. G. Rittenhouse, ‘‘Phishing attacks and defenses,'' Int. J. Secur. Appl., vol. 10, no. 1, pp. 247–256, 2016.

[5] M. Hijji and G. Alam, ‘‘A multivocal literature review on growing social engineering based cyber-attacks/threats during the COVID-19 pandemic: Challenges and prospective solutions,'' IEEE Access, vol. 9, pp. 7152–7169, 2021.

[6] A. Alzahrani, ‘‘Coronavirus social engineering attacks: Issues and

recommendations," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 5, pp. 154–161, 2020.

[7] Srikanth, V. "Secret Sharing Algorithm Implementation on Single to Multi Cloud." Srikanth | International Journal of Research, 23 Feb. 2018, journals.pen2print.org/index.php/ijr/article/view/11641/11021.

[8] V. Srikanth. "Managing Mass-Mailing System in Distributed Environment" v srikanth | International Journal & Magazine of Engineering, Technology, Management and Research, 23 August. 2015. http://www.ijmetmr.com/olaugust2015/VSrikanth-119.pdf

[9] V. Srikanth. "SECURITY, CONTROL AND ACCESS ON IOT AND ITS THINGS" v srikanth | INTERNATIONAL JOURNAL OF MERGING TECHNOLOGY AND ADVANCED RESEARCH IN COMPUTING, 15 JUNE. 2017. http://ijmtarc.in/Papers/Current%20Papers/IJMTARC-170605.pdf

[10] V. Srikanth. "ANALYZING THE TWEETS AND DETECT TRAFFIC FROM TWITTER ANALYSIS" v srikanth | INTERNATIONAL JOURNAL OF MERGING TECHNOLOGY AND ADVANCED RESEARCH IN COMPUTING, 20 MARCH. 2017. http://ijmtarc.in/Papers/Current%20Papers/IJMTARC-170309.pdf

[11] V. Srikanth. "A NOVEL METHOD FOR BUG DETECTION TECHNIQUES USING INSTANCE SELECTION AND FEATURE SELECTION" v srikanth | INTERNATIONAL JOURNAL OF INNOVATIVE ENGINEERING AND MANAGEMENT RESEARCH, 08 DECEMBER. 2017. https://www.ijiemr.org/public/uploads/paper/976_approvedpaper.pdf

[12] V. Srikanth. "SECURED RANKED KEYWORD SEARCH OVER ENCRYPTED DATA ON CLOUD" v srikanth | INTERNATIONAL JOURNAL OF INNOVATIVE ENGINEERING AND MANAGEMENT RESEARCH, 08 Febraury. 2018. http://www.ijiemr.org/downloads.php?vol=Volume-7&issue=ISSUE-02

[13] V. Srikanth. "WIRELESS SECURITY PROTOCOLS (WEP,WPA,WPA2 & WPA3)" v srikanth | Journal of Emerging Technologies and Innovative Research (JETIR), 08 mAY. 2019. https://www.jetir.org/papers/JETIRDA06001.pdf

[14] V. Srikanth, et al. "Detection of Fake Currency Using Machine Learning Models." Deleted Journal, no. 41, Dec. 2023, pp. 31–38. https://doi.org/10.55529/ijrise.41.31.38.

[15] V. Srikanth, et al. "A REVIEW ON MODELING AND PREDICTING OF CYBER HACKING BREACHES." 25 Mar. 2023, pp. 300–305. http://ijte.uk/archive/2023/A-REVIEW-

ON-MODELING-AND-PREDICTING-OF-CYBER-HACKING-BREACHES.pdf.

[16] V. Srikanth, “DETECTION OF PLAGIARISM USING ARTIFICIAL NEURAL NETWORKS.” 25 Mar. 2023, pp. 201–209. http://ijte.uk/archive/2023/DETECTION-OF-PLAGIARISM-USING-ARTIFICIAL-NEURAL-NETWORKS.pdf.

[17] V. Srikanth, “CHRONIC KIDNEY DISEASE PREDICTION USING MACHINELEARNINGALGORITHMS.” 25 January. 2023, pp. 106–122. http://ijte.uk/archive/2023/CHRONIC-KIDNEY-DISEASE-PREDICTION-USING-MACHINE-LEARNING-ALGORITHMS.pdf.

[18] Srikanth veldandi, et al. “View of Classification of SARS Cov-2 and Non-SARS Cov-2 Pneumonia Using CNN”. journal.hmjournals.com/index.php/JPDMHD/article/view/3406/2798.

[19] Srikanth veldandi, et al. “Improving Product Marketing by Predicting Early Reviewers on E-Commerce Websites.” Deleted Journal, no. 43, Apr. 2024, pp. 17–25. https://doi.org/10.55529/ijrise.43.17.25.

[20] Srikanth veldandi, et al.“Intelligents Traffic Light Controller for Ambulance.” Journal of Image Processing and Intelligent Remote Sensing, no. 34, July 2023, pp. 19–26. https://doi.org/10.55529/jipirs.34.19.26.

[21] Veldandi Srikanth, et al. “Identification of Plant Leaf Disease Using CNN and Image Processing.” Journal of Image Processing and Intelligent Remote Sensing, June 2024, https://doi.org/10.55529/jipirs.44.1.10.

[22] Y. Cao, W. Han, and Y. Le, ‘‘Anti-phishing based on automated individual white-list,’’ in Proc. 4th ACM Workshop Digit. Identity Manage. (DIM), 2008, pp. 51–59.

[23] P. Prakash, M. Kumar, R. R. Kompella, and M. Gupta, ‘‘PhishNet: Predictive blacklisting to detect phishing attacks,’’ in Proc. IEEE INFOCOM, Mar. 2010, pp. 1–5.

[24] A. K. Jain and B. B. Gupta, ‘‘A novel approach to protect against phishing attacks at client side using auto-updated white-list,’’ EURASIP J. Inf. Secur., vol. 2016, no. 1, pp. 1–11, Dec. 2016.

[25] E. Buber, B. Diri, and O. K. Sahingoz, ‘‘NLP based phishing attack detection from URLs,’’ in Proc. Int. Conf. Intell. Syst. Design Appl., vol. 736, 2018, pp. 608–618.

AUTHOR’SPROFILE

Ms.V. SAVITHRI currently she is working as Assistant Professor inAudisankara College of Engineering andTechnology (AUTONOMOUS),NH-5,BypassRoad,Gudur,Tirupati(Dt.),Andhra Pradesh, India.

V.SURENDRAis pursuing MCA from Audisankara College of Engineering and Technology (AUTONOMOUS), NH-5, Bypass Road, Gudur, Tirupati (Dt.), Andhra Pradesh, India.