Comparative Study of CatBoost, XGBoost, and LightGBM for Enhanced URL Phishing Detection: A Performance Assessment

Ammar Odeh[1*], Qasem Abu Al-Haija[2], Abdullah Aref[3] and Anas Abu Taleb[4]

[1*] Department of Computer Science, Princess Sumaya University of Technology, Jordan.
a.odeh@psut.edu.jo, Orcid: https://orcid.org/0000-0002-9929-2116

[2] Department of Cybersecurity, Princess Sumaya University of Technology, Jordan.
q.abualhaija@psut.edu.jo, Orcid: https://orcid.org/0000-0003-2422-0297

[3] Department of Data Science, Princess Sumaya University of Technology, Jordan.
a.aref@psut.edu.jo, Orcid: https://orcid.org/0000-0001-5172-7179

[4] Department of Computer Science, Princess Sumaya University of Technology, Jordan.
a.abutaleb@psut.edu.jo, Orcid: https://orcid.org/0000-0002-8286-1829

Received: July 12, 2023; Accepted: September 20, 2023; Published: November 30, 2023

Abstract

Phishing via URLs involves cyber attackers crafting deceptive websites or emails, mimicking genuine entities like banks or social media outlets. The objective is to dupe users into divulging personal data, such as their passwords or card numbers. This study assesses the potential of machine learning in identifying phishing domains by constructing and contrasting three distinct models. These models, crafted using CatBoost, XGBoost, and LightGBM techniques, are then juxtaposed against prior solutions documented in academic literature. We employed the UCI phishing domains dataset, sourced from URLs, as a performance benchmark for our models. Findings indicate that the model built on CatBoost outperforms its counterparts and also surpasses earlier documented methods.

Keywords: CatBoost, XGBoost, LightGBM, Social Engineering.

1 Introduction

Social engineering attacks are fraudulent acts that aim at manipulating individuals and enterprises to disclose sensitive data. Such attacks can be classified as social-based, where relationships with victims are used; physical-based, where attackers perform physical actions to reveal the desired data; or technical-based, where information technology is used to reveal the desired data (Sigala, M., 2019). URL Phishing attacks, also known as web phishing, are widely used technology-based social engineering attacks. A phishing URL is a website address designed to look like a legitimate website to trick victims. Such attacks are increasingly being launched via email, text, social media, gaming, or dating applications (Nguyen, G., 2019). Phishing approaches can generally be classified into five categories: whitelist-, blacklist-, content-, visual similarity- and URL based (Shorten, C., 2019). Recently, machine learning has been successfully used for URL phishing detection.

Journal of Internet Services and Information Security (JISIS), volume: 13, number: 4 (November), pp. 01-11.
DOI: *10.58346/JISIS.2023.I4.001*

*Corresponding author: Department of Computer Science, Princess Sumaya University of Technology, Jordan.

URL Phishing has been around for more than 30 years, and more victims are deceived yearly, especially during the COVID-19 pandemic (Vinayakumar, R., 2019). Common tactics for URL Phishing include link manipulation, using images instead of text to avoid phishing detection filters, malicious use of web scripting languages (such as Javascript) to hide the actual browser address bar, and misusing browser vulnerabilities.

The literature suggests numerous phishing mitigation strategies, often introducing an additional security layer during user login attempts (Sivaraman, K., 2019). Implementing these strategies can be challenging due to the potential need for modifications on the website, collaboration, the risk of complicating user interfaces, or incurring additional authentication computation costs (Dwivedi, A.D., 2019). Another avenue is user education, equipping individuals to recognize phishing attempts. Yet, relying solely on this method proves suboptimal, especially with the prevalence of inexperienced users (Al-Turjman, F., 2019). In contrast, phishing detection methods consistently outshine both prevention techniques and user training (Sivaraman, K., 2019). These detection methods can operate on the client side, or through specific software on the host or server side (Kumar, S., 2018). Importantly, they often demand minimal user training and bypass the need to alter existing authentication processes (Ang, L.M., 2018).

Phishing has grown exponentially over the past few years, with reports indicating that over 1.5 million new phishing sites are created monthly. In 2020 alone, an estimated $3.5 billion was lost to phishing scams, highlighting the situation's urgency. For instance, in a high-profile case last year, a prominent company fell victim to a phishing attack that compromised the data of over 200,000 users. The financial and reputational costs associated with such breaches can be staggering. Therefore, employing machine learning to detect these malicious domains becomes not just beneficial but crucial.

Conventional methods of phishing detection often resort to white or blacklisting, where legitimate websites are cataloged on a whitelist, while confirmed phishing sites are added to a blacklist for widespread dissemination, safeguarding users from potential threats. Nonetheless, these methods falter when confronted with novel phishing URLs, leaving unsuspecting users vulnerable until the malicious link is identified and blacklisted (Lau, B.P.L., 2019). Another tactic involves rule-based detection, where security experts derive rules from URL components, such as domain name similarities to recognized legitimate domains (Wu, Y., 2019). However, as phishers become privy to these rules, they devise strategies to circumvent them (Mosavi, A., 2019).

Phishing detection has been modelled as a classification, and supervised machine learning has been used successfully for detecting phishing websites with various algorithms being tested (such as naïve Nayes, logistic regression, decision tree, and random forest) aiming to achieve a high accuracy and low false warning rate (Palanisamy, V., 2019).

2 Literature Review

Basit et al. employed a balanced website phishing dataset from the UCI machine learning repository with 11055 records and 30 features; the ensemble algorithm of K-NN combined with RFC attained an accuracy of 97.3% and the f-measure of 0.976, which was the best among all other models. The proposed classification method involved using the random forest classifier and combining it with three different classifiers (ANN, C4.5, and K-NN) using ensemble majority voting (Sadowski, J., 2019).

XGBoost, Random Forests, Bagging, Adaboost, LightGBM, and Gradient Boost were among the genetic algorithm-based ensemble classifiers suggested for use in Al-Sarem et al.'s (Saura, J.R., 2019) paper, which also used three medium-sized website phishing datasets (1000–31000 variables in each dataset), with datasets 1 and 3 showing some imbalance and dataset 2 being balanced. Random Forests produced the best results in the first and third datasets with 97.02% and 97.15% accuracy and f-measures of 0.9749 and 0.9590, respectively. LightGBM achieved the best accuracy, precision, and recall in the second dataset, reporting a 98.65% accuracy rate and an f-measure of 0.9865.

Authors in (Nallaperuma, D., 2019) suggest a lightweight deep learning system. Experimental tests and comparisons have confirmed the recommended method's effectiveness. The experiments show an increase in the correct detection rate. The proposed method's ability to operate in real-time on an embedded single-board computer with energy-saving features has also been confirmed in this work.

In the approach outlined in (Schulz, S., 2019), the authors methodically reduced the feature set to discern which URL attributes are pivotal for identifying phishing sites, all while maintaining high accuracy. This investigation utilized two datasets containing 48 and 87 features respectively. In the first dataset, a fusion of power predictive score correlation and recursive feature elimination was applied. The second dataset employed maximal information coefficient correlation alongside recursive feature elimination, while the third scenario integrated recursive feature elimination with Spearman correlation. Impressively, across all scenarios, even when leveraging the smallest feature subset, the combined methodologies consistently yielded high accuracy. Specifically, with just 10 features, Dataset 1 achieved an accuracy of 97.06%, while Dataset 2 reported a 95.88% accuracy.

In (Shang, C., 2019), Using a character-level convolutional neural network (CNN) for phishing detection based on the URL of the website, authors suggested a quick deep learning-based solution model. The proposed model does not call for using any third-party services or retrieving any content from the target website. Without needing prior phishing knowledge, it collects data and sequential patterns of URL strings and uses those attributes to classify the URL quickly. Comparisons between numerous classical machine learning models and deep learning models are offered for evaluations utilizing a range of feature sets, including hand-crafted, character embedding, character level TF-IDF, and character level count vector features. The proposed model outperformed the current phishing URL models in the experiments, achieving an accuracy of 95.02% on our dataset and 98.58%, 95.46%, and 95.22% on benchmark datasets.

In (Yu, Y., 2019), authors examine certain common characteristics displayed by phishing websites and create a model to identify these websites. Several models, including Random Forest Classifier, Decision Tree Classifier, Logistic Regression, K Nearest Neighbors, Artificial Neural Networks, and Max Vote Classifier of Random Forest, Artificial Neural Networks, and K Nearest Neighbors, were trained on the dataset. The Max Vote Classifier of Random Forest (max depth 16), Decision Tree (max depth 18), and Artificial Neural Network (97.73%) had the highest accuracy.

The goal of the authors in (Huang, M., 2019) was to train a system using Artificial Neural Networks (ANNs) and Deep Neural Networks (DNNs) based methods to detect odd requests by analyzing the URLs of web pages. We trained the algorithm using a dataset with 36,400 trustworthy web pages and 37,175 phishing attempts. The experimental results show that the suggested approaches, which employ ANN and DNN approaches, respectively, have an accuracy rate of 92% and 96% in detecting phishing websites.

A machine learning-based anti-phishing solution, known as PHISH-SAFE, was proposed by the author in (Xu, G., 2019) and is based on Uniform Resource Locators (URLs) properties. We have taken 14 elements from the URL to determine whether a website is phishing or not in order to assess the performance of our suggested solution. More than 33,000 authentic and phishing URLs were used to train the recommended system utilizing SVM and Naive Bayes classifiers. According to the findings of our investigation, an SVM classifier can detect phishing websites with 92% accuracy.

3 Method

Research Hypotheses:

H1: Machine learning models can effectively detect phishing domains with higher accuracy than traditional methods.

H2: Among the three evaluated machine learning models, there will be significant differences in their effectiveness in detecting phishing domains.

Research Questions:

RQ1: How influential are the three machine learning models in detecting phishing domains compared to traditional methods?

RQ2: Which of the three machine learning models has the highest accuracy in detecting phishing domains?

RQ3: What are the key features or factors the most effective machine learning model uses to distinguish between legitimate and phishing domains?

This paper aims to develop an ensemble machine-learning model that can classify URLs as phishing or benign (safe) ones (Johnson, C., 2020). The proposed system is supposed to take any URL as input and accurately predict the class. In this research, we aim to find the effectiveness of using three different gradient-boosting algorithms: CatBoost, XGBoost and LightGBM.

Our system architecture consisted of three main stages: 1-data preparation, data collection, data cleaning, and feature engineering. 2-learning process: training our proposed ensemble models on the training subset of the data, parameter tuning, and validation. 3-evaluation using our evaluation metrics.

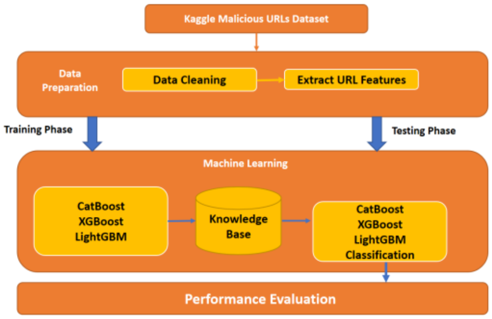

Figure 1: Framework for the Proposed System

The dataset used in this research was the Kaggle malicious URLs dataset (Aqib, M., 2019), which included 522214 URLs; the counts of the URLs belonging to each class are shown in the table below.

Table 1: Data Classification

	Benign	Phishing
Train	342488	75283
Test	85615	18828
Total	428103	94111

The dataset included no features. In order to make it easier for the model to detect patterns more efficiently, we performed feature engineering. The final dataset consisted of 18 engineered features.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{1}$$

r= correlation coefficient

x_i= values of the x-variable in a sample

$\bar{x}$=mean of the values of the x-variable

y_i=values of the y-variable in a sample

$\bar{y}$=mean of the values of the y-variable

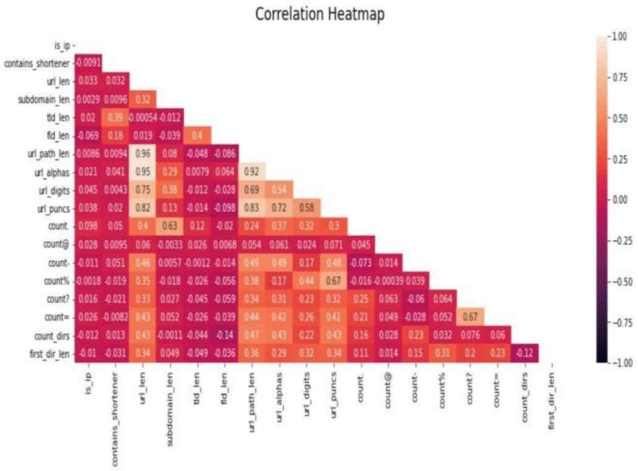

Figure 2: Pearson Correlation Heatmap

Following data preparation, the dataset was divided into training (80%) and testing (20%) subsets following the Pareto 80/20 rule. The three models that were suggested were all gradient-boosting techniques.

As a unit test, a single decision tree model was created to compare it to the ensemble model and see if it was successful and performed better than the single model.

Boosting is an ensemble machine learning technique that employs decision trees one at a time and gradually combines them to improve predictions from the model.

Due to their capacity to learn from the errors of each previous model by assigning weights to each prior prediction, boosting algorithms were used. Additionally, boosting provides a quick and accurate solution to categorization issues. The three algorithms employed were:

CatBoost

CatBoost is known for its ability to handle categorical features effectively without requiring manual encoding or preprocessing. It also uses various techniques to prevent overfitting, such as random permutations of feature values and gradient-based sampling of the training data (Leonelli, S., 2020). First, we used the default hyperparameters of the model. Then, after using grid search and cross-validation to find the best hyperparameters for the problem, we set the learning rate to 0.1 max depth to 10, meaning that each decision tree will have a maximum depth of 10.

Hyperparameter Tuning: A grid search was performed using the CatBoost built-in hyperparameter tuning tool. The key hyperparameters tuned included learning_rate, depth, l2_leaf_reg, and iterations.

Feature Engineering: Given CatBoost's ability to handle categorical data directly, nominal variables like 'domain_suffix' were passed without one-hot encoding. Numeric features were scaled using Min-Max scaling.

XGBoost

XGBoost is known for its efficiency and scalability, and it has been used in many winning solutions in machine learning competitions. It can handle various data types, including numerical and categorical data, and supports classification and regression tasks (Stylos, N., 2019).

Hyperparameter Tuning: A combination of random search and grid search was used for hyperparameter tuning using Scikit-learn's GridSearchCV. Hyperparameters tuned included learning_rate, max_depth, min_child_weight, gamma, subsample, and colsample_bytree.

Feature Engineering: Categorical variables were one-hot encoded. Numeric features were standardized using Z-score normalization. Feature importance was assessed using the plot_importance function from XGBoost, and non-influential features were removed.

LightGBM

LightGBM (Light Gradient Boosting Machine) is a gradient boosting framework that Microsoft developed. It is an open-source, distributed machine-learning library designed to be fast, scalable, and efficient(Song, Q., 2019).

LightGBM supports various data types, including categorical features, and can be used for classification and regression tasks. It also includes several techniques to prevent overfitting, such as early stopping and regularization. A 5-fold cross-validation reduces bias and ensures the model performs well with any subset of testing data.

Hyperparameter Tuning: Bayesian optimization was applied for hyperparameter tuning using the Optuna library. The key hyperparameters tuned included learning_rate, num_leaves, min_data_in_leaf, feature_fraction, and bagging_fraction.

Feature Engineering: Categorical variables were label encoded. Numeric features were scaled using Robust scaling. Based on the feature importance plot from LightGBM, some feature interactions were created to enhance the model's predictive power.

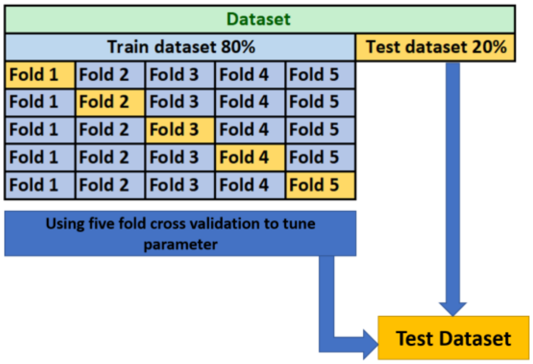
Dataset
Train dataset 80%
Test dataset 20%
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Fold 1 Fold 2 Fold 3 Fold 4 Fold 5
Using five fold cross validation to tune parameter
Test Dataset

Figure 3: Visual Representation of a 5-fold Cross-validation Process

After training the model, binary classification will be carried out on the testing data, and the model's performance will be evaluated using the following evaluation metrics: accuracy, precision, and recall using the confusion matrix.

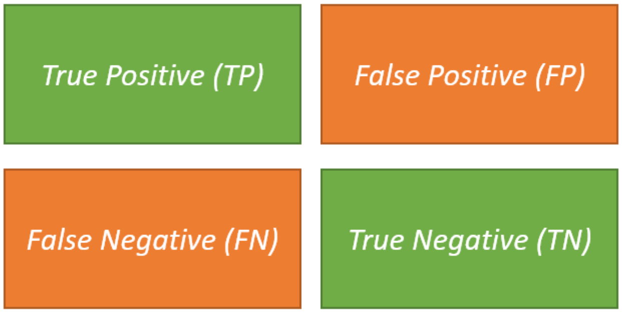

Figure 4: Confusion Matrix

True Positive (TP) refers to accurately classifying a sample into the positive class.

True Negative (TN) refers to a sample being accurately identified as being in the negative class.

False Positive (FP) refers to a sample that should have been classed as positive but was instead incorrectly assigned to the negative class.

False Negative (FN) refers to a sample that should be categorised as positive but is mistakenly placed in the negative class.

The measures we used are calculated using the following equations.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (2)$$

$$Precision = \frac{TP}{TP + FP} \qquad (3)$$

$$Recall = \frac{TP}{TP + FN} \qquad (4)$$

4 Results and Discussion

This section explains the research results and, at the same time, gives. Our research revealed that the ensemble model performed better than the traditional techniques. The single model used was a decision tree model, and it achieved an accuracy of 95%, an f-measure of 0.97, a recall of 0.97, and a precision of 0.97.

In this work, we developed an ensemble machine-learning model for detecting whether a URL is benign or phishing. To measure the effectiveness of the proposed approach, we have evaluated the system performance in terms of the evaluation metrics mentioned above, and we report our empirical results in this section.

The table above shows that the CatBoost model was the best-proposed model with an accuracy of 96.9% and an F-measure of 0.98, so this model can be selected from this research to predict and detect phishing websites.

Figure 5 below shows the feature importance of the CatBoost model.

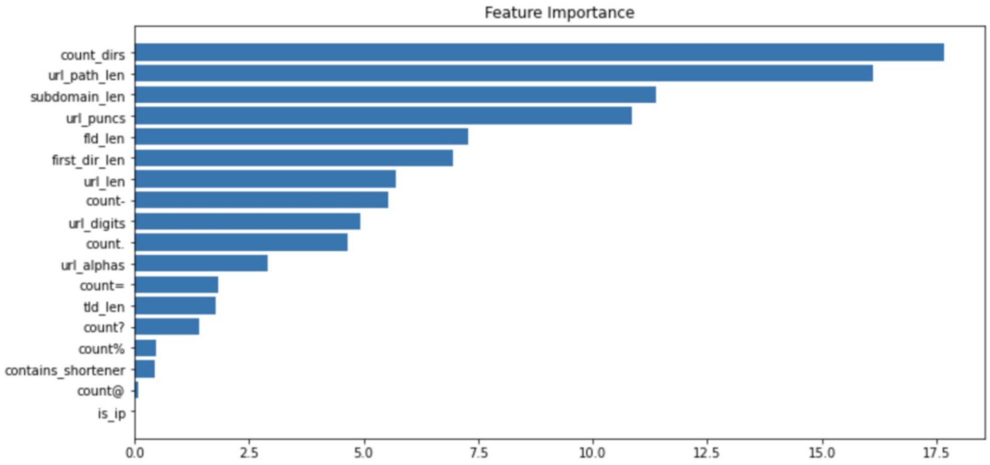

Figure 5: Features Importance

Table 2: Performance Summary of Proposed Models

	CatBoost	XGBoost	Light GBM
Precision	0.98	0.93	0.96
Recall	0.98	0.97	0.98
Accuracy	96.9%	92.1%	95.2%
F-measure	0.98	0.95	0.97

Table 3: Comparison of Related Works and Proposed Algorithm

Ref	Algorithm	F-measure	Accuracy
[14]	ANN, C4.5, and K-NN	98%	97%
[15]	Random Forests	96%	97%
[18]	CNN	95%	95%
[19]	DTC, LR	94%	97%
[20]	ANN, DNN	93%	92%
[21]	SVM and Naive Bayes	92%	92%
Proposed	**CatBoost**	**98%**	**97%**
	XGBoost	95%	92%
	LGBM	97%	95%

The class imbalance problem is solved using the F-measure score as our primary metric to assess a model's performance. For instance, if the class ratio is 99:1 and the model consistently predict the first class, it will achieve 99% accuracy without considering the second class.

5 Conclusion

In conclusion, three models were created and evaluated in this paper's proposal to apply gradient-boosting learning in detecting phishing websites. After completing feature engineering, the proposed models (CatBoost, XGBoost, and LGBM) were evaluated on the Kaggle harmful URLs dataset. The models' task was categorising the URLs as benign or phishing in binary form. We looked at many different applications for these models to get the best results. We tried to make them more effective by using grid search for parameter tweaking and cross-validation to lessen model bias.

The accuracy for the CatBoost, XGBoost, and LGBM models was 96.9%, 92.1%, and 95.2%, respectively, with f-measures of 0.98, 0.95, and 0.97. Overall, the CatBoost model outperformed the other suggested models in this research.

Given that the dataset we used for our research was significantly more significant than any of the cited relevant works, our proposed model performed well compared to other research that used boosting. The future development of a comprehensive system with improved accuracy and less bias is our key objective. We want to test our best model on other datasets to assess it further and enhance its overall performance. Additionally, the system can be expanded and refined to label particular URLs as questionable websites, alerting users and lowering the rate of false negatives.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Ammar Odeh: Conceptualization, methodology, data analysis, and writing of the manuscript. Qasem Abu Al-Haija: Data collection, literature review, and manuscript revision. Abdullah Aref: Methodology development, experimental design, and manuscript editing. Anas Abu Taleb: Software implementation, data interpretation, and manuscript proofreading

Acknowledgements

The authors sincerely acknowledge the Princess Sumaya University for Technology for supporting steps of this work.

References

[1] Sigala, M., Beer, A., Hodgson, L., & O'Connor, A. (2019). Big data for measuring the impact of tourism economic development programmes: A process and quality criteria framework for using big data. *Big Data and Innovation in Tourism, Travel, and Hospitality: Managerial Approaches, Techniques, and Applications*, 57-73.

[2] Nguyen, G., Dlugolinsky, S., Bobák, M., Tran, V., López García, Á., Heredia, I., & Hluchý, L. (2019). Machine learning and deep learning frameworks and libraries for large-scale data mining: a survey. *Artificial Intelligence Review*, *52*, 77-124.

[3] Shorten, C., & Khoshgoftaar, T.M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, *6*(1), 1-48.

[4] Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. *Ieee Access*, *7*, 41525-41550.

[5] Sivaraman, K., Krishnan, R.M.V., Sundarraj, B., & Sri Gowthem, S. (2019). Network failure detection and diagnosis by analyzing syslog and SNS data: Applying big data analysis to network operations. *International Journal of Innovative Technology and Exploring Engineering*, *8*(9) Special Issue 3, 883-887.

[6] Dwivedi, A.D., Srivastava, G., Dhar, S., & Singh, R. (2019). A decentralized privacy-preserving healthcare blockchain for IoT. *Sensors*, *19*(2), 1-17.

[7] Al-Turjman, F., Zahmatkesh, H., & Mostarda, L. (2019). Quantifying uncertainty in internet of medical things and big-data services using intelligence and deep learning. *IEEE Access*, *7*, 115749-115759.

[8] Kumar, S., & Singh, M. (2018). Big data analytics for healthcare industry: impact, applications, and tools. *Big data mining and analytics*, *2*(1), 48-57.

[9] Ang, L.M., Seng, K.P., Ijemaru, G.K., & Zungeru, A.M. (2018). Deployment of IoV for smart cities: Applications, architecture, and challenges. *IEEE access*, *7*, 6473-6492.

[10] Lau, B.P.L., Marakkalage, S.H., Zhou, Y., Hassan, N.U., Yuen, C., Zhang, M., & Tan, U.X. (2019). A survey of data fusion in smart city applications. *Information Fusion*, *52*, 357-374.

[11] Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., & Fu, Y. (2019). Large scale incremental learning. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 374-382.

[12] Mosavi, A., Shamshirband, S., Salwana, E., Chau, K.W., & Tah, J.H. (2019). Prediction of multi-inputs bubble column reactor using a novel hybrid model of computational fluid dynamics and machine learning. *Engineering Applications of Computational Fluid Mechanics*, *13*(1), 482-492.

[13] Palanisamy, V., & Thirunavukarasu, R. (2019). Implications of big data analytics in developing healthcare frameworks–A review. *Journal of King Saud University-Computer and Information Sciences*, *31*(4), 415-425.

[14] Sadowski, J. (2019). When data is capital: Datafication, accumulation, and extraction. *Big data & society*, *6*(1), 1-12.

[15] Saura, J.R., Herráez, B.R., & Reyes-Menendez, A. (2019). Comparing a traditional approach for financial Brand Communication Analysis with a Big Data Analytics technique. *IEEE access*, *7,* 37100-37108.

[16] Nallaperuma, D., Nawaratne, R., Bandaragoda, T., Adikari, A., Nguyen, S., Kempitiya, T., & Pothuhera, D. (2019). Online incremental machine learning platform for big data-driven smart traffic management. *IEEE Transactions on Intelligent Transportation Systems*, *20*(12), 4679-4690.

[17] Schulz, S., Becker, M., Groseclose, M.R., Schadt, S., & Hopf, C. (2019). Advanced MALDI mass spectrometry imaging in pharmaceutical research and drug development. *Current opinion in biotechnology*, *55*, 51-59.

[18] Shang, C., & You, F. (2019). Data analytics and machine learning for smart process manufacturing: Recent advances and perspectives in the big data era. *Engineering*, *5*(6), 1010-1016.

[19] Yu, Y., Li, M., Liu, L., Li, Y., & Wang, J. (2019). Clinical big data and deep learning: Applications, challenges, and future outlooks. *Big Data Mining and Analytics*, *2*(4), 288-305.

[20] Huang, M., Liu, W., Wang, T., Song, H., Li, X., & Liu, A. (2019). A queuing delay utilization scheme for on-path service aggregation in services-oriented computing networks. *IEEE Access*, *7,* 23816-23833.

[21] Xu, G., Shi, Y., Sun, X., & Shen, W. (2019). Internet of things in marine environment monitoring: A review. *Sensors*, *19*(7), 1-21.

[22] Aqib, M., Mehmood, R., Alzahrani, A., Katib, I., Albeshri, A., & Altowaijri, S.M. (2019). Smarter traffic prediction using big data, in-memory computing, deep learning and GPUs. *Sensors*, *19*(9), 1-34.

[23] Leonelli, S., & Tempini, N. (2020). *Data journeys in the sciences*, 412. Springer Nature.

[24] Stylos, N., & Zwiegelaar, J. (2019). *Big data as a game changer: how does it shape business intelligence within a tourism and hospitality industry context?*, 163-181. Springer Singapore.

[25] Song, Q., Ge, H., Caverlee, J., & Hu, X. (2019). Tensor completion algorithms in big data analytics. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, *13*(1), 1-48.

[26] Johnson, C., Khadka, B., Basnet, R.B., & Doleck, T. (2020). Towards Detecting and Classifying Malicious URLs Using Deep Learning. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 11*(4), 31-48.

Authors Biography

Ammar Odeh received his Ph.D. Degree in Computer Science and Engineering with a concentration in Computer Security (Steganography) from the University of Bridgeport. He received an M.S. in Computer Science with a concentration in Reverse Software Engineering and Computer Security from the University of Jordan, College of King Abdullah II School for Information Technology (KASIT). In 2002, he finished his B.Sc. Degree in Computer Science and applications from the Hashemite University, Prince Al-Hussein Bin Abdullah II for Information Technology. During his Ph.D., he worked as a Research Assistant, Teaching Assistant, and Instructor. He is currently an assistant professor in computer science at Princess Sumaya University for Technology.

Qasem Abu Alhija received his Ph.D. from Tennessee State University (TSU), USA, in 2020. He is an Assistant Professor at the Department of Cybersecurity, School of Computing Sciences, Princess Sumaya University for Technology (PSUT), Amman, Jordan. He authorised more than 100 scientific research papers and book chapters. His research interests include Artificial Intelligence (AI), Cybersecurity and Cryptography, the Internet of Things (IoT), Cyber-Physical Systems (CPS), Time Series Analysis (TSA), and Computer Arithmetic.

Abdullah M. Aref received his Ph.D. from the School of Electrical Engineering and Computer Science at the University of Ottawa in May 2018. In September 2018, he joined Princess Sumaya University for Technology as an Assistant Professor in the Computer Science Depts. Since October 2022, he has been a member of the Data Science department. His research interests include Trust Management, Multi-agent systems, Machine Learning, and NLP.

Anas Abu Taleb is an associate professor in the Department of Computer Science at Princess Sumaya University for Technology, Amman, Jordan. He received a Ph.D. in Computer Science from the University of Bristol, UK, in 2010, an MS.c. in Computer Science from the University of the West of England, UK, 2007 and a BS.c. Degree in Computer Science from Princess Sumaya University for Technology, Jordan, 2004. Dr. Abu Taleb has published several journal and conference papers on sensor networks. In addition to sensor networks, Dr. Abu Taleb is interested in network fault tolerance, routing algorithms, and mobility models.