Towards Trust-based Data Weighting in Machine Learning

Seán Óg Murphy
School of Computer Science & Information Technology
University College Cork
Cork, Ireland
seanogmurphy@ucc.ie

Utz Roedig
School of Computer Science & Information Technology
University College Cork
Cork, Ireland
u.roedig@cs.ucc.ie

Cormac J. Sreenan
School of Computer Science & Information Technology
University College Cork
Cork, Ireland
cjs@cs.ucc.ie

Ahmed Khalid
Dell Research
Dell Technologies
Cork, Ireland
ahmed.khalid@dell.com

***Abstract*—In distributed environments, data for Machine Learning (ML) applications may be generated from numerous sources and devices, and traverse a cloud-edge continuum via a variety of protocols, using multiple security schemes and equipment types. While ML models typically benefit from using large training sets, not all data can be equally trusted. In this work, we examine data trust as a factor in creating ML models, and explore an approach using annotated trust metadata to contribute to data weighting in generating ML models. We assess the feasibility of this approach using well-known datasets for both linear regression and classification problems, demonstrating the benefit of including trust as a factor when using heterogeneous datasets. We discuss the potential benefits of this approach, and the opportunity it presents for improved data utilisation and processing.**

***Index Terms*—edge computing, machine learning, data confidence fabric, linear regression, clustering, data weighting**

I. Introduction

In a distributed cloud-edge continuum, data emanates from a multiplicity of sources that have heterogeneous physical and logical characteristics. Devices may have hardware with different security levels; data producers may or may not encrypt data as it traverses the network towards the cloud; data may pass through parts of the network with different security levels. Data might be stored in secure, encrypted data storage or a less secure alternative medium.

Data may be generated by formally accredited or authenticated contributors, or otherwise. The sensors or devices generating the data may or may not have been installed by certified engineers, or at the time of their installation use security technologies with different characteristics. These uncertainties lead to lack of trust and confidence in decentralized environments with multiple stakeholders acting independently, using diverse mechanisms or standards [1].

A rising area in network security is that of Zero Trust architectures [2], where, rather than implicitly trusting data at any part of the network chain, it is evaluated at the edge near where it is produced, as it traverses the network, also centrally or in the cloud, and where it is consumed. Incorporating trust evaluation schemes and trust validation frameworks in distributed environments can enable trustworthy communication among entities with low or zero trust.

Trust evaluation frameworks allow for data to be treated differently based on its provenance. For example, Project Alvarium [3], enables informed decision-making for applications in distributed environments, by assigning confidence scores to data as it passes along a cloud-edge continuum. Similar employment of confidence or trust scores to artificial intelligence and machine learning (AI/ML) applications, may lead to better-performing models by weighting data according to its trustworthiness.

Typically, data weighting in AI/ML applications is performed to balance an imbalanced dataset [4]. For instance a particular data class may be over-represented in the dataset and this can be compensated for by weighing members of that class proportionately lower than the other classes of data. We may also be interested in particular classes more than others and hence may wish to weigh those data entries more heavily during the training process.

In this work, we explore the idea of weighing data when training ML models, using a trust score metric as would be provided through a Trust-based framework. This trust score is used to determine how much we should trust a given datapoint based on its provenance and handling. The key contribution of this work is to establish whether weighing data based on its trustworthiness can improve the performance of AI/ML models. Through experiments, we validate our hypothesis, and identify the extent and scenarios where trust-based data

This work was conducted as part of the CLEVER project, EU Grant Number 101097560 and EI No: IR-2022-0065, and supported in part by a research grant from Science Foundation Ireland (SFI), Grant Number 13/RC/2077_P2.

weighting benefits AI/ML models.

In the remainder of this paper, we start by defining the concept of data trust along with some background material and related work (Section II). We then explain the ML techniques that are evaluated in our experiments (Section II-C). The core of the paper is a set of experiments (Section III) using two well known datasets, that allow us to evaluate (Section IV) the feasibility of using Data Trust and explore its potential. We provide a conclusion (Section V) and potential future work (Section VI) that shows how our paper can act as a seed for follow-on research that recognises the critical role that data security must play in training dependable ML models.

II. Related Work

Recent industry trends towards Edge computing have given rise to decentralized and distributed environments, such as cloud-edge continuum. Consequently, the concept of a Zero Trust approach to networking systems has come to prominence [2]. In this paradigm, data may be generated and traverse a network but is not explicitly trusted by data consumers. To allow for safe and secure handling of data, and to ensure that the processing results can be trusted, knowledge about the data is used with a "Trust Algorithm": safety/security policies that determine how the data was used, transmitted, stored, destroyed, and so on.

One approach to building trust in such environments is through the use of data annotation, i.e. metadata that is mapped to data traversing the network, which is used by Trust Algorithms to evaluate how to deal with it. These annotations can include the security class of the device that generated that data, the qualifications or identities of the person who installed that device, its encryption state, the security classes of network infrastructure that data has passed through so far, whether it has been stored in immutable and secure storage, whether it has been registered onto a ledger, and so on. One approach is to implement this functionality as a Data Confidence Framework (DCF).

A. Data Confidence Frameworks

The proliferation of distributed decentralized environments, including multi-Edge multi-cloud continua, has introduced new challenges in trust and confidence for organizations and stakeholders. In such environments, multiple data producers generate data, using resources that are spread across various locations and providers, leading to complex trust relationships and potential vulnerabilities. Trust and confidence become critical concerns due to the lack of centralized control, increased attack surface, and the involvement of multiple independent stakeholders [1].

Distributed Ledger Technologies (DLTs), such as Hashgraphs [5] or blockchains, implement distributed consensus algorithms to generate immutable logs and enable decentralized trust verification among multiple parties. DLTs can be used to build frameworks where communication among applications can be carried out with a certain level of trust and confidence in an otherwise low or zero-trust environment. E.g., An information sharing scheme has been proposed proposed [6] that uses smart contracts to calculate and track reputation of participants and filter out untrustworthy information.

DTMS [7] proposed a trust management framework that deploys an evaluation model and uses blockchain to create an irreversible storage of trust credits. Project Alvarium [3] goes beyond state-of-the-art by creating a generic framework that builds a data confidence fabric by: tracking data events as the data flows across a cloud-edge continuum; recording associated metadata and annotations in a DLT; calculating confidence scores based on the trust insertion technologies used, and; delivering data to applications with measurable confidence.

In the context of AI/ML applications measurable confidence, provided by frameworks such as Alvarium, can be used when sharing and exchanging data sets for training AI models. Because, the quality of data may impact the effectiveness and reliability of AI models, incorporating confidence scores into the data exchange process, can lead to improved performance in decentralized environments with multiple independent data producers. Such a framework will be of key interest in solutions that rely on federated learning or swarm intelligence [8], due to their inherent distributed decentralized nature.

B. Mixed Trust

As mentioned above, AI/ML datasets for a given problem may come from a mix of sources, of varying quality and provenance. Similarly, survey results may come from volunteers, censuses etc. and data collection might be from a mix of private enterprise sensors with publicly-submitted data. Consequently, Data confidence frameworks (Section II-A), feature a mix of different security annotations that are considered differently by Trust Algorithms, reflecting the relative amount of trust being placed in those data points. For example, Regeru et al [9] considered the importance of mixed provenance and quality data in health settings, where datapoints are reported by health workers, finding that depending on the reporting source and supervisors, the quality of data reporting varied substantially and quality supervision was a key aspect. Also in the health domain, Searle et al. [10] explored the challenges and considerations for patient confidentiality in IoT networks, including the potential use of Zero Trust architectures to preserve privacy and maintain data quality. While Machine Learning applications sometimes consider relative data quality as a factor, this is used in order to filter and sanitise datasets, discarding low-quality material – in our work we continue to make use of the lower-trust material but diminish its contribution to models via weighting. In our work we employ a "data poisoning" approach for introducing lower quality material into the training sets, and make use of trust scores to mitigate their impact, this is similar to the work of Venkatesan et al [11], but differing in that their approach is to use several trained models and to use their disagreements as a method to mitigate the impact of poisoned data.

In this work we consider the consequences and possible advantages in accounting for this trust scoring and using it

as a source of weighting in ML computation. An important component in Data Confidence Fabric/Zero Trust systems is the use of a "Trust Algorithm" which uses the given metadata to determine how it should be trusted and treated. Machine Learning has been explored to drive Trust algorithms both centrally and at the network edge [12] [13] [14] [15]. While in our work we use Trust scores as an input for ML models, the quality of model output based on this weighting can be useful as an input into such Trust algorithm models, as we discuss in Section VI.

C. Machine Learning

To evaluate the potential advantages of accounting for Trust scores in networks, we make use of two ML techniques. We use Random Forest Regression to solve a numerical problem predicting housing values. For a classification problem, that of predicting the employment category of a customer, we use Decision Trees. Our experiments (Section III) were carried out using scikit-learn [16].

1) Random Forest Regression: Linear Regression techniques [17] are a family of ML approaches for numerically continuous modelling: predicting a numerical value based on input fields. Performance is considered typically in terms of Mean Squared Error (MSQ) or the coefficient of determination (R^2) value [18]. In this work we use a Random Forest Regression approach [19] for predicting house values (Section III-A) – this approach uses a number of different regression trees each trained on subsets of the training data, with their predictions averaged to improve precision and avoid overfitting of the data.

2) Decision Tree Classification: Classification techniques are used to predict field values that are discrete and may or not be ordered – given a particular input, the model can predict a particular class for that input. Decision Tree Classification is an ML approach that examines a field at a time and splits the dataset into two branches above or below a given value, or True and False for a given class. Certain paths in the tree can be used to make predictions as to remaining field values further down the tree. While this can be used for linear regression problems, it is also useful for classification of a given field, which we use in predicting the job type of correspondents to a bank marketing campaign (Section III-B). Where there isn't a linear relationship between certain fields and the classification target, or where certain fields have unordered possible values, this technique is well-suited.

III. Methodology

Different ML techniques traditionally allow for data weighting based on particular objects – a common issue with regression and classification problems is that datasets may be imbalanced, certain classes may be over- or under-represented which can cause models trained on that data to be skewed based on their relative representation in the inputs. Some implementations of regression or classification techniques allow for individual fields to be assigned a weight value, and entries are then weighted either explicitly according to that value, or relative based on the ordering of the weights.

Another method for weighting data is oversampling or undersampling data based on how much it should contribute to the subsequent model. In this approach, entries that we wish to contribute more are duplicated (or entries for which we wish to diminish their impact have a reduced chance to contribute to the model, a chance to be discarded).

While these weighting approaches are most commonly used for dealing with data with imbalanced entries for which we are more interest in some entries than others, in this work we are weighting data based on its Trust score evaluation as would be provided by Trust Algorithms in a Zero Trust context (such as through a Data Confidence Framework).

In this work, we perform two sets of experiments by taking existing datasets, Housing (Section III-A) and Banking (Section III-B), and conduct a Machine Learning task on each (Random Forest Regression and Decision Tree Classification respectively). The datasets are broken out into an 80%/20% Training/Testing split – models are trained on 80% of the dataset and evaluated using the remaining 20%.

The initial results of these tasks are evaluated based on their accuracy in predicting values for entries within their respective Testing sets, providing a Baseline result (labeled B). For the Housing (regression) task results are evaluated by their R^2 value; values closer to 1.0 indicate better results. For the Banking (classification) task, the results are evaluated according to the percentage accuracy of the classifications – that is to say the percentage of the Testing set which was accurately classified; in this case a higher percentage indicates a better result.

The next step is to alter a 10% proportion of entries within the Training set ("data poisoning") and conduct the experiments again. This is repeated with 20%, 30% data poisoning up to 90% and these results are evaluated (the "Poisoned" results). The process is repeated but in this next set of experiments the data entries which have been "poisoned" are excluded from training the model – this gives a progressively smaller training set for each iteration as we are discarding 10, 20, 30% of the training data each time. These are the "Clean" results. Finally we repeat the poisoning operation but this time each entry within the poisoned portion of the dataset is labeled with an low Trust score (a value of 1 in this work), while the remaining data has a Trust score of 10. The Trust scores are used to weight the data in the Training set according to how much it is trusted in the ML models (Sections III-A and III-B). These are the "Trust" results. Values of 1 and 10 were chosen to focus our work on demonstrating the principles of this concept in action. The standard data weighting for the ML approaches used considers weights based on their relative ordering rather than absolute values, the values of 10 and 1 reflect the suggested Trust scoring mechanism of the Project Alvarium (which typically scores Trust as a value between 1 and 10). In future work we will explore using a continuum of Trust values and a more varied range of data quality and confidence and the consequences of a range of

weighting in ML models, where there may be data scored with an intermediate trust score between these two extremes.

A. California Housing Prices Dataset

The California Housing Prices Dataset is based on census data for the state of California, USA collected in 1990. This dataset contains median values for housing blocks across the state, which each entry representing a single housing block. The fields in the dataset include ocean proximity, house median age, total number of rooms in the housing block and so on, and this dataset was chosen due to it's popularity in Machine Learning studies and education (Géron [20], Nugent [21]). For these experiments, the fields used are 'longitude', 'latitude', 'median house value' and 'median income'.

As one might expect, there is a correlation between the median income for a housing block and the median house value for that block. There is also a weaker correlation between the latitude of housing with the house value, as California is elongated north-to-south and features relatively expensive coastal cities interspersed with national park areas and agricultural land.

In these experiments, the task is to predict the median house value of a datapoint from the Testing set based on the values of the other fields above. The Data Poisoning procedure for the regression experiments using the housing data is to take a portion of the dataset (in 10, 20, 30, ..., 90 % increments) and within this portion each entry has a 50% chance to have its latitude manipulated northward by 2 degrees – this having the effect of moving what would be upper-end urban residences into low-priced areas between major coastal cities which we would expect to make the housing price prediction poor when tested with the unaltered test set.

B. Portuguese Bank Marketing Dataset

The Portuguese Bank Marketing Dataset (Moro et al [22]) is the result of a direct phone marketing campaign carried out by a Portuguese banking institution between 2008 and 2010. Entries in this dataset include the age of the recipient, their bank balance, education level etc. and include a yes/no field for whether the recipient decided to avail of the financial product being offered by the bank. For these experiments, we used the fields of 'age' (in years), 'marital' (single, married or divorced), 'education' (primary, secondary, tertiary), 'balance' (in euros) and 'job' (management, entrepreneur, retired, technician, admin, services, blue-collar, self-employed, student or unemployed).

In these experiments, we use a Decision Tree model to predict the job type of each entry in the Testing set based on the values of the fields above. The Data Poisoning procedure in these classification experiments is to change the job type in the training set to a random value – this poisoning worsens the accuracy of the model when used on the test set.

IV. Experiment Results

In our experiments using Random Forest Regression for predicting median house value on the California Housing Data

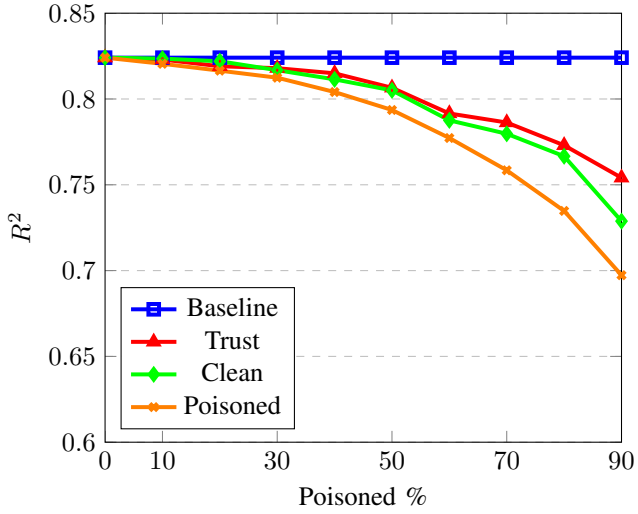

Fig. 1. Housing Random Forest Regression results

(Figure 1), we attained a baseline R^2 result of 0.824, this being the quality of the trained model using the fully unmodified 80% Training set. In our Clean result, we find that the quality of prediction decays as we discard further and further portions of the dataset due to it being part of the "poisoned" set – with less data to train on, the model misses out on some insights. When the poisoned data is included with the clean data, the model also performs poorly, giving progressively poorer results as more of the training set has been manipulated. Finally we observe that when the Trust weighting is accounted for, a mix of clean and poisoned data can actually give an improved result over the Clean dataset alone – having more data to train on has improved the model while the impact of the manipulated data's impact on the model has been tempered through the Trust weighting. Important explanatory information within the dataset has been able to contribute insights to the model even though its provenance merits circumspection.

In the Banking Classifcation experiments, the baseline accuracy for predicting the "job" value using Decision Trees is 41.12%. In our Clean result, we find a relatively consistent decline in performance as the training set becomes smaller, as was found with the Housing experiments. The training set containing the mix of altered and unaltered data ("Poisoned") performs the least well at the outset, and declines sharply as a greater proportion of the set contains poisoned data. When accounting for Trust as a weighting input for the models ("Trust"), even when containing manipulated entries, the prediction performance is generally the best of the three. In comparison with the Linear Regression problem for the Housing experiments, the variation in accuracy results is somewhat coarser. As a classification problem based on a mix of field labels and numerical values, there is much less of a linear relationship between the predictor fields and the predicted output.

We find that for datasets where a small proportion of the set

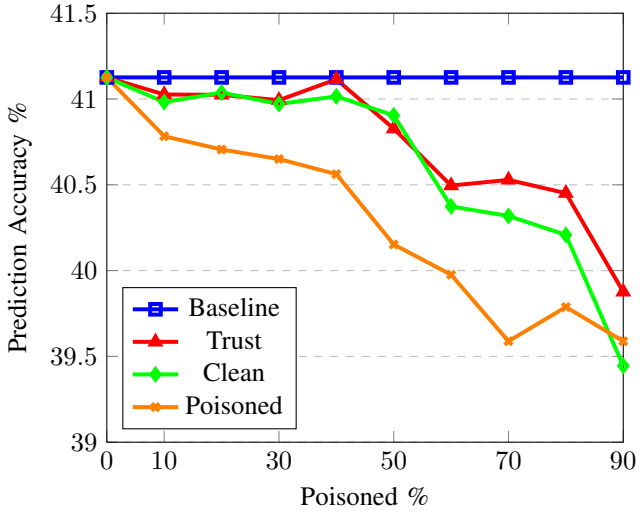

Fig. 2. Banking Decision Tree Classification results

is of lower trust, there is little to no penalty from disregarding the lower-trust material – if there is sufficient volume of unaltered/highly trusted data then the input is representative of the overall problem and the model performs well on the test set.

As a larger proportion is found to be low-quality or due to low-security regimes is liable to be manipulated, the potential for poor model outcomes increases – using only the maximally trusted material leads to poor results as the training set is smaller and omits potentially important information and it is no longer representative of the sort of data points possible. Similarly, does treating all data equally even if some has be altered leads to poor results as the manipulation impacts on the model and its predictions. As the test set is unaltered, the training material is significantly different in its character for the tampered field and hence the prediction results suffer.

In our experiments we found that it is possible to achieve strong results with data of mixed provenance provided that the data is appropriately annotated so that relative Trust scores can be assigned to each entry and then these are used in generating ML model. This allows for a large volume of data to feed the model, while dampening the contribution of the low-trust material. While the "poisoned" data has been modified, there is still useful explanatory information within it (particularly the unaltered fields), and hence the mixed trust models continue to perform well even for high volumes of altered entries.

V. Conclusions

We introduced the concept of Data Trust as an input for Machine Learning tasks; in systems and applications where data may be produced by a variety of sources and arrive through a variety of vectors, Trust evaluation can be used from the edge to the cloud to determine how that data is processed, forwarded, and consumed. For Machine Learning applications within such an environment where there is a relatively high volume of data generated but the trust evaluation of any individual datum can vary, we investigated ML data weighting techniques to determine if there was merit in considering mixed Trust data in driving ML models as opposed to considering all inputs of equivalent weighting, or discarding the less-than-ideal input material. While we used only two values for Trust (1 or 10) in these proof of concept experiments, Data Confidence Fabrics and other Trust Platforms support a continuum of trust values, which would allow for a range of weightings to be used in models – in future we will explore the impact of adjusting this range of weighting for more varied trust metrics.

In some deployments, it may be the case that there is no meaningful difference between the models generated by the entirety of data sources vs. those generated by just the maximally trusted sources, but in this work we have demonstrated that if there is a meaningful qualitative difference between one part of the dataset and the rest, that we might still avail of that lower quality information if it still provides explanatory power in models. In future work (Section VI) we will examine when and where to determine if initially untrusted data should have been trusted – variable trust weighting for trust annotations based on model evaluations.

The results of the experiments conducted in this work demonstrate that trust-based data weighting for training AI/ML models can improve the performance of the models. This leads to a variety of interesting questions and research opportunities for machine learning applications that run in distributed environments.

VI. Future Work

While some algorithms exist in literature for calculating trust score of data, research needs to be carried out to design or determine optimal methods for calculating trust scores and weights for AI/ML models. Such investigation could involve exploring different trust metrics, incorporating contextual information, evaluating the impact of trust scores on the performance of the model, or leveraging external knowledge sources to enhance the accuracy of trust estimation.

Furthermore, in this work we used pre-computed trust scores for data points based on the known chance that a given data point might have been manipulated. In future experiments we are exploring trust scoring based on DCF annotations in live deployments where the trust annotations for data points are passed to trust algorithms for computing trust values and driving the trust-based weighting in machine learning.

We used fixed trust scores in our experiments, where data from the suspect portion of the training sets has a fixed low trust score and the known unaltered material has the maximum score – in a live deployment we would expect a range of trust scoring but also we would need to investigate the operation of the trust algorithms themselves.

For a given data processing (including network forwarding) or machine learning task we may need to consider a variety of trust criteria respective to the intended use of the data. Some data may be so poorly trusted that it isn't worth the overhead to propagate it throughout a network, whereas sometimes data

may be of medium trust quality but still useful for analysis or ML tasks of intermediate sensitivity.

In related work, we identified research activities that explore the possibility of establishing a score value for certain annotations based on its performance within ML models – trust criteria scoring that can be learned. This approach may also be of use in anomaly detection, where a vulnerability introduced due to a particular Trust feature being missing for some data can be identified when that Trust feature is weighted higher or lower and hence has a negative impact on results.

Applying trust-based data weighting to real-world AI/ML applications across different domains can, generate further insights, enable online adaptation, and assess deployment feasibility and scalability. Within the context of CLEVER project [23], trust-based data weighting will be evaluated in various use-cases over a collaborative cloud-edge continuum, where federated learning is used for AI/ML applications in heterogeneous distributed environments.

-

REFERENCES

[1] A. A. Monrat, O. Schelén, and K. Andersson, "A survey of blockchain from the perspectives of applications, challenges, and opportunities," *IEEE Access*, vol. 7, pp. 117 134–117 151, 2019.

[2] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," National Institute of Standards and Technology, Tech. Rep., 2020.

[3] Oct 2021. [Online]. Available: https://www.lfedge.org/projects/alvarium/

[4] H. Kaur, H. S. Pannu, and A. K. Malhi, "A systematic review on imbalanced data challenges in machine learning: Applications and solutions," *ACM Computing Surveys (CSUR)*, vol. 52, no. 4, pp. 1–36, 2019.

[5] L. Baird, "The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance," *Swirlds Tech Reports SWIRLDS-TR-2016-01, Tech. Rep*, vol. 34, pp. 9–11, 2016.

[6] Y. Liu, X. Hao, W. Ren, R. Xiong, T. Zhu, K.-K. R. Choo, and G. Min, "A blockchain-based decentralized, fair and authenticated information sharing scheme in zero trust internet-of-things," *IEEE Transactions on Computers*, vol. 72, no. 2, pp. 501–512, 2022.

[7] X. Chen, J. Ding, and Z. Lu, "A decentralized trust management system for intelligent transportation environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 558–571, 2020.

[8] M. A. Ağca, S. Faye, and D. Khadraoui, "A survey on trusted distributed artificial intelligence," *IEEE Access*, vol. 10, pp. 55 308–55 337, 2022.

[9] R. N. Regeru, K. Chikaphupha, M. Bruce Kumar, L. Otiso, and M. Taegtmeyer, "'do you trust those data?'—a mixed-methods study assessing the quality of data reported by community health workers in kenya and malawi," *Health Policy and Planning*, vol. 35, no. 3, pp. 334–345, 2020.

[10] R. Searle and P. Gururaj, "Protecting patient confidentiality in the internet of medical things through confidential computing," *Journal of Data Protection & Privacy*, vol. 5, no. 4, pp. 347–362, 2023.

[11] S. Venkatesan, H. Sikka, R. Izmailov, R. Chadha, A. Oprea, and M. J. De Lucia, "Poisoning attacks and data sanitization mitigations for machine learning models in network intrusion detection systems," in *MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*. IEEE, 2021, pp. 874–879.

[12] A. Hatamian, M. B. Tavakoli, M. Moradkhani *et al.*, "Improving the security and confidentiality in the internet of medical things based on edge computing using clustering," *Computational Intelligence and Neuroscience*, vol. 2021, 2021.

[13] J. Byabazaire, G. O'Hare, and D. Delaney, "Data quality and trust: Review of challenges and opportunities for data sharing in iot," *Electronics*, vol. 9, no. 12, p. 2083, 2020.

[14] J. Byabazaire, G. M. O'Hare, R. Collier, and D. Delaney, "Iot data quality assessment framework using adaptive weighted estimation fusion," *Sensors*, vol. 23, no. 13, p. 5993, 2023.

[15] S. K. Gallagher, A. Whisnant, A. D. Hristozov, and A. Vasudevan, "Reviewing the role of machine learning and artificial intelligence for remote attestation in 5g+ networks," in *2022 IEEE Future Networks World Forum (FNWF)*. IEEE, 2022, pp. 602–607.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[17] S. Weisberg, *Applied linear regression*. John Wiley & Sons, 2005, vol. 528.

[18] S. A. Glantz, B. K. Slinker, and T. B. Neilands, *Primer of applied regression & analysis of variance, ed*. McGraw-Hill, Inc., New York, 2001, vol. 654.

[19] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[20] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.

[21] C. Nugent, "California housing prices," Nov 2017. [Online]. Available: https://www.kaggle.com/datasets/camnugent/california-housing-prices

[22] S. Moro, R. Laureano, and P. Cortez, "Using data mining for bank direct marketing: An application of the crisp-dm methodology," 2011.

[23] "CLEVER Project." [Online]. Available: https://www.cleverproject.eu/