An Ensemble Based Computational Social System for Fake News Detection in MANET Messaging

Amit Neil Ramkissoon (amit.ramkissoon@my.uwi.edu)

The University of the West Indies Saint Augustine Campus: The University of the West Indies at St Augustine https://orcid.org/0000-0003-2164-3366

Wayne Goodridge

The University of the West Indies Saint Augustine Campus: The University of the West Indies at St Augustine

Research Article

Keywords: Computational Social System, Content, Credibility, Ensemble Learning, Fake News Detection, MANET

Posted Date: January 25th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1208481/v1

An Ensemble Based Computational Social System for Fake News Detection in MANET Messaging

Amit Neil Ramkissoon

Department of Computing & Information Technology

The University of the West Indies at St Augustine

St Augustine, Trinidad & Tobago

amit.ramkissoon@my.uwi.edu

ORCID ID: 0000-0003-2164-3366

Wayne Goodridge, PhD

Department of Computing & Information Technology

The University of the West Indies at St Augustine

St Augustine, Trinidad & Tobago

wayne.goodridge@sta.uwi.edu

All authors contributed to the study conception and design. Conceptualization, Methodology, Software, Data curation, Writing- Original draft preparation, Visualization and Investigation were performed by Amit Neil Ramkissoon. Supervision, Writing- Reviewing and Editing were performed by Wayne Goodridge, PhD. All authors read and approved the final manuscript.

The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

An Ensemble Based Computational Social System for Fake News Detection in MANET Messaging

Abstract— Mobile Adhoc Networks (MANETs) are utilised in a variety of mission critical situations and as such it is important to detect any fake news that exists in such networks. This research combines the power of Veracity, a unique, computational social system with that of Legitimacy, a dedicated ensemble learning technique, to detect Fake News in MANET Messaging. Veracity uses five algorithms namely, VerifyNews, CompareText, PredictCred, CredScore and EyeTruth for the capture, computation and analysis of the credibility and content data features using computational social intelligence. To validate Veracity, a dataset of publisher credibility-based and message content-based features is generated to predict fake news. To analyse the data features, Legitimacy, a unique ensemble learning prediction model is used. Four analytical methodologies are used to analyse these experimental results. The analysis of the results reports a good performance of the Veracity architecture combined with the Legitimacy model for the task of fake news detection in MANET Messaging.

Keywords— Computational Social System, Content, Credibility, Ensemble Learning, Fake News Detection, MANET

1. Statements and Declarations

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. On behalf of all authors, the corresponding author states that there is no conflict of interest.

2. Introduction

Given the mission critical nature of Mobile Ad hoc Networks (MANETs), it is essential to predict fake news in its messages. According to (Murugan and Shanmugam, 2012) Mobile Ad Hoc Networks (MANETs) can be defined a collection of mobile devices that are dynamic, independent, and wireless and connected together to form a communications network. This communication network is divorced of any infrastructure and operates solely amongst the members. The network is considered as self-configuring as its topology and behaviour changes to meet the dynamic structure of the network. In MANETs each member operates as a publisher, a subscriber, and a router at some point in time during its involvement in the network.

MANETs are used in a variety of applications especially in a social context by way of message sharing. MANET messaging has become an integral part of today's social communication landscape as it is used in a variety of applications. As stated in (Stieglitz and Fuchß, 2011) the social and commercial benefits of MANETs surpass the technical issues associated with them, making any deployment of MANET technology for mobile applications a success.

Predicting and detecting fake news is a challenge. Many consumers of online news especially via social media and internet enabled platforms do not cross reference news posted to such websites and as such are unable to verify the credibility of the news. Hence automated credibility validation of news is a necessity (Khan et al., 2019).

Fake news is defined as any news article that is intentionally and verifiably false (Shu et al., 2017). In recent times, the spread of fake news has become one of the newest issues the world has had to face as seen in the COVID-19 infodemic (Pian, Chi, and Ma, 2021). The problem has been further exacerbated by the use of social media platforms and communication networks for message exchange. As stated in (Lazer et al., 2018) allowing fake news to spread especially via social media and internet enabled platforms erodes the long-standing efforts that have been made and achieved against the spread of misinformation. Also, allowing this spread which is international, allows for mistruths, misinformation, and spurious conspiracies to establish a living ecosystem. One recent example of the danger of the spread of fake news is a tweet posted by American rapper Nikki Minaj that indicated false information about the Covid-19 vaccines. This tweet however was retweeted 117,700 times with Ms Minaj having a visibility of 22.1 million followers leading to international news headlines, scientific investigations into this false claim (O. Blackstock and U. Blackstock, 2021) and a negative spotlight on Trinidad and Tobago.

A major methodology used for the prediction fake news is the analysis the features of fake news to determine if any relationships exist amongst these features. According to (Zhou and Zafarani, 2018) fake news detection is

subdivided into four categories based upon the perspective of the detection strategy. These perspectives as stated in (Zhou and Zafarani, 2018) are (i) knowledge-based, which focuses on the knowledge in the published news and fact checks it to determine whether it is fake or not; (ii) style-based, which focuses on how the published news is written and checks for subtle similarities with the styles of both genuine and fake news; (iii) propagation-based, which focuses on the travel pattern of the published news and how it spreads, and (iv) credibility-based, which investigates the validity of the published news based upon the credibility of the publisher and spreaders.

As stated above credibility based fake news detection focuses on detection techniques based upon the established reputation of the news publisher. Credibility based detection, a part of context-based detection, pays particular attention to the credibility of the news publisher. If the news publisher is seen as a person of ill-repute, then the news cannot be trusted and vice versa. According to (Zhou and Zafarani, 2018), when attempting to detect fake news based upon the credibility of the publisher, the detection involves data surrounding the online social behaviour of the publisher as well as the news-related information. For example, a message published by an unreliable publisher and forwarded by unreliable network clients is less likely to be trustworthy and more likely to be fake news than news posted by authoritative and credible users.

Alongside credibility based fake news detection, content based fake news detection, a sub-genre of style-based detection is also utilised to identify fake news. As stated in (X. Zhang and Ghorbani, 2020) fake news is comprised of both the physical and non-physical content. The physical content constitutes the features of the news like title, body text, image, or video, whilst the non-physical content constitutes the likes of purpose, sentiment, and news topics. From a content perspective the language, structure, syntax, and expression of written fake news can be investigated and compared to that of the wider topic. As stated in (Zhou and Zafarani, 2018) all of the textual features of the news can be investigated via, content-based detection. One major problem that computational social systems can be applied to is the credibility and content detection of the spread of fake news in Mobile Ad Hoc Network (MANET) Messaging.

One of the major social computing issues that affect message sharing via Mobile Ad Hoc Networks (MANETs) is that of the spread of fake news. This problem is having serious deleterious effects on our social data driven society. As stated in (Nazir et al., 2016) in MANETs, one class of attacks is known as the fabrication attack whereby the attacker sends fake news to its neighbours in the network.

To accomplish the task of fake news detection, this work proposes the ensemble learning based Veracity architecture. The Veracity architecture was introduced by (Ramkissoon & Goodridge, 2021). Veracity investigates how does a publisher's social behaviour relates to the legitimacy of his content. Veracity is a combined credibility and content based multidimensional social computing architecture for fake news detection that focuses on the capture, computation, and analysis of news publisher credibility on MANETs. As such many features of the users must be investigated to understand how they indicate credibility of the news. The Veracity architecture attempts to model social behaviour and human reactions to news spread over a MANET by using computational social systems. The Veracity architecture works in a fully distributed and infrastructureless environment.

To gather these features five algorithms are introduced by the Veracity architecture namely, VerifyNews, CompareText, PredictCred, CredScore and EyeTruth. Each of these algorithms assist in the capture, computation and analysis of the credibility and content- based data of messages shared on the MANET. This research builds upon the established Veracity architecture by enhancing the predictability of the architecture. To accomplish the enhancement, Veracity utilizes credibility-based and content-based features and the Legitimacy ensemble learning (ML) model to predict whether news is fake or not.

Legitimacy is a unique ensemble learning model for the task of Credibility based Fake News Detection. This model consists of two underlying techniques namely, a Two Class Boosted Decision Tree and a Two Class Neural Network. The Legitimacy model follows a pseudo mixture-of-experts methodology of combining ML techniques. To accomplish the functionality of a gating model, Logistic Regression is implemented to combine to output of the two main models.

As stated above credibility features can be further subdivided into distinct perspectives of features. As such (Ramkissoon & Goodridge, 2021) proposed two further subdivisions of credibility-based features. The credibility-based features are further subdivided into the demographic and online social behaviour subdivisions. As the name suggests, the demographic features look at the data features that relates to personal traits of an individual e.g. Created At, Language, Location, Screen Name, Time Zone, User Language. These features remain constant in

most messages and as such do not change. The online social behaviour features, relate to features of the user and how he/she has interacted on the social network. These features relate to how the network perceives him/her. These features include Credibility Score, Eyewitness Score, Eyewitness, Favourites Count, Followers Count, Friends Count, Label, Listed Count, Source, Statuses Count, The Text and Text Similarity. According to (Ramkissoon & Goodridge, 2021) the Two Class Boosted Decision Tree works best with the online social behaviour features whilst the Two Class Neural Network works best with the demographic features. Hence, the ensemble combines the results from both methods to provide for better data analysis.

This work proposes an ensemble based computational social systems approach to the problem of fake news detection in MANET messaging. In an effort to model social behaviour to understand human intuition, a unique field of research named computational social systems has emerged. According to (University RWTH Aachen, 2021) our society is being digitalized at an unstoppable rate, which is presenting us with new challenges. An increasing amount of data are being recorded with new sensors and AI systems. This digitalization process lets us measure individual, organizational, and societal behaviour with unprecedented resolution and very high precision. The availability of these new data forms requires individuals, organizations, and society to grapple with the effects of digitalization in our everyday lives, social systems, and the larger society. This development offers great economic and social opportunities on the one hand, yet also poses a multitude of problems for society – making it necessary to continuously develop our understanding of complex social and digital systems. It raises questions of the responsible usage of personal data, the control of data, the development of a comprehensive data ethic, the social responsibility for data and algorithms, and our trust in so-called computational social systems. Computational social systems are systems in which social and societal processes, procedures, and functions are digitally reproduced and formed through the dynamic interaction of algorithms and social behaviour.

Given the relationship between social behaviour and machine learning, this work merges both these fields, and proposes the Ensemble-Based Veracity Architecture and the five associated algorithms, as a computational social system. This system employs computational social systems, using an ensemble model, in an attempt to detect fake news in MANET messaging.

This paper proposes the following:

1. To present the Ensemble-Based Veracity architecture for fake news detection in MANET messaging
2. To discuss the VerifyNews, PredictCred, CredScore, CompareText and EyeTruth algorithms
3. To discuss the Legitimacy ensemble learning model
4. To propose the combine Ensemble-Based Veracity Architecture
5. To present and evaluate initial results of the model for fake news detection

The remainder of this paper is structured as follows. Section 2 presents the related work in this field whilst Section 3 presents an understanding of the Veracity architecture and the VerifyNews, PredictCred, CredScore, CompareText and EyeTruth algorithms. Experiments are conducted in Section 6 using the algorithms and the results of these are discussed in Section 7. Section 8 concludes the paper.

3. Literature Review

Prior research in fake news detection for MANET message sharing has been very sparse. Most of the prior research efforts have focused on Fake News Detection in the MANET subcategory of Vehicular Ad Hoc Networks (VANETs).

According to (Sohail et al., 2019), there is a need for trusted information sharing in future vehicular networks to provide a platform for road safety and news sharing. This type of platform however also provides an opportunity for malicious users to share and disseminate fake news to the network. Solutions such as Traditional Public Key Infrastructure do not provide sufficient protection against these malicious users as these users are all authorized entities. To address this problem their work provides the Three-Valued Subjective Logic (3VSL) as a solution. This solution operates as a trust model for multi-hop trust assessment among the clients in Vehicular Ad Hoc Networks (VANET). Vehicle users provide their opinions on news via the 3VSL, and these opinions are stored as instances of trust. These instances of trust are updated frequently due to the random movement of vehicles on the roadways. Thy support their hypothesis with two types of simulations i.e., numerical, and experimental analysis. The numerical analysis shows that the 3VSL provides accurate trust assessment in any network topology.

These topologies include bridge and random topologies, topologies that were previously ignored due to edge splitting. In the experimental analysis, they extended AODV (Ad-hoc On Demand Vector), the well-known ad hoc routing protocol, by improving the routing tables to include trust fields. The analysis of the simulations prove that their proposed model achieves good performance with low mobility VANETs but was poor when compared with high mobility VANETs with respective to throughput and latency. Their system utilises roadside units and trust calculations. It ignores the knowledge and credibility of the message which can help to further identify the fake news. Finally, their system allows the message to spread before it can conclude that the news is fake.

According to (Daza et al., 2008), the sharing of vehicle-generated announcements is amplified by the use of vehicular ad hoc networks (VANETs) since they allow vehicle to- vehicle communication. These announcements can be beneficial to users of the network since they can increase the probability of driving safely, provided that these announcements can be trusted. To this end they presented in their work a secure system for vehicle-generated announcements. This new system proposed to protect against the dissemination of fake news messages from both internal and external attackers. To protect against internal attacks, they proposed an endorsement mechanism based on threshold signatures. Based on their experiments and results they concluded that their system outperformed previous security schemes in terms of message length and computational cost. To ensure that users did not have to forego their privacy in order to endorse messages they also described three different privacy preserving variants of the system.

As stated in (Xiao, Liu, and Li, 2020), road safety and traffic efficiency in the Internet of Vehicles (IoV) is severely impacted by the unabated spread of fake news. Hence it is important to identify fake news in a timely fashion, given the nature of the IoV. To this end they proposed Quick Fake News Detection (QcFND) in their paper, which operates as a network computing framework. This framework utilises the technologies of Software-Defined Networking (SDN), edge computing, blockchain, and Bayesian networks to protect against fake news. QcFND consists of at two levels: the network's edge and the vehicles themselves. At the edge of the network are a number of Software-Defined Road Side Units (SDRSUs), which are built as an extension of the traditional Road Side Units (RSUs). These RSUs host SND controllers and blockchain serves to which the vehicles connect and are built as virtual machines. The SDN controllers are used to implement the load balancing on IoV. At the blockchain servers the fake news opinion reports submitted by vehicles are collected and analysed. From this analysis the probability of the presence of a traffic event is calculated, in an effort to provide timely updates to the passing vehicles. The Bayesian Networks are utilised to infer whether to trust the received traffic reports. To validate the performance of their proposed work, they tested the QcFND's performance three different platforms namely, Veins, Hyperledger Fabric, and Netica. The analysis of this experimental results show that QcFND achieved good performance as compared with others. Their scheme requires a constant connection to the roadside units and infrastructure network for fake news detection as detection is done at a central location.

An example of computational social intelligence at work is provided by (Liang, 2020c) which states that presently enormous amounts of information is generated on the Internet in the field of emergency management research especially in public social platforms and this information is not being fully utilised. Information like the precious scene images sent by the parties risking their lives may be one of the victims of this under-utilisation and as such they precious value may be underestimated. In other words, they state that present systems may not be equipped enough to consider this valuable information that deal with unconventional emergencies for the purposes of early warnings and monitoring. Building upon this foundation and the failures of emergency information collection and transmission their chapter an effective image acquisition and emergency treatment solution for unconventional emergencies. This solution is built to complement the information source of emergency management field decision making. Their solution raises the profile of the need for greater emphasis to be placed on the field of emergency decision making.

According to another similar work proposed by (Liang, 2020b) when news is posted to a network in the area of automatic reading and decision support systems, one the task executed is the automatic segmentation of words. This automatic segmentation automatically partitions the words into various pieces. The news is then compared to the previously established keyword library in the form of vectors. The classifier then outputs the news category based upon the words and the training based upon historical data. When news that has not been analysed by the classifier before enters the system, the keywords database, historical news database, and historical transaction database are updated, and the support vector machine classifier is trained again.

Prior research in fake news detection and machine learning has been conducted. According to (Collins et al., 2020) an overview of the various models in detecting fake news such as Machine learning, Natural Language Processing,

Crowdsourced techniques, Expert fact-checker, as well as Hybrid Expert-Machine are introduced. The aforementioned research also examined differing types of fake news, which is an essential criterion for detection. The findings show that detecting fake news is a challenging but workable task. The techniques which combine people and machines bring very satisfactory results. It is further stated that Early Machine Learning methods in detecting fake news assume fake news is created intentionally for the political and financial benefit, so they often have an opinionated and enticing headline, as such the extraction of the textual and linguistic feature is necessary for ML. Naive Bayes classifier and classified linguistic features such as lexical features, including word count and level, as well as syntactic nature, which involves sentence level characterization are used.

As stated in (Shu et al., 2018) Model-oriented fake news research opens the door to building more effective and practical models for fake news detection. Most previously mentioned approaches focus on extracting various features, incorporating these features into supervised classification models, such as naive Bayes, decision tree, logistic regression, k nearest neighbour (KNN), and support vector machines (SVM), and then selecting the classifier that performs the best. More research can be done to build more complex and effective models and to better utilise extracted features, such as aggregation methods, probabilistic methods, ensemble methods, or projection methods.

According to (Khan et al., 2021) an overall performance analysis of different approaches on three different datasets is presented. It is shown that Naive Bayes with n-gram can attain analogous results to neural network-based models on a dataset with less than 100k news articles. The performance of Long Short-Term Memory (LSTM) based models greatly depends on the length of the dataset as well as information given in a news article. With adequate information provided in a news article, LSTM based models have a higher probability to overcome overfitting. Moreover, advanced models like Convolutional neural network (CNN) LSTM (C-LSTM), Conv HAN and character level C-LSTM have shown high promise that demands further attention on these models in fake news detection. Finally, a topic-based analysis that exposes the difficulty to detect political, health and research-related deceptive news is performed.

As illustrated in (Gaonkar et al., 2019) several models are utilised to detect fake news online. Four different models to detect fake news are mainly utilised. Naïve Bayes, Support Vector Machine (SVM), Logistic Regression and Multilayer Perceptron methodologies to detect fake news are utilised. An SVM is applied to a dataset consisting of 12600 truthful articles and 12600 fake articles. First, the dataset is pre-processed using stop word removal and stemming features are extracted by using term frequency and term frequency–inverse document frequency (TF-IDF) and a feature matrix is formed from the documents. This is then passed through a classifier. The classifier consists of six different machine learning algorithms, Stochastic Gradient Descent (SGD), SVM, Linear Support Vector Machine (LSVM), KNN and Decision Trees (DT). The highest accuracy was obtained when using unigrams features and linear SVM giving an accuracy of 92%.

As reported in (Zhang & Ghorbani, 2020) Supervised machine learning algorithms like DT, Random Forest, SVM, Logistic Regression, KNN are extensively used in previous literatures for online hoaxes, frauds, and deceptive information classification.

As stated in (Ahmad et al., 2020) Ensemble Learning has been used previously in Fake News Detection. According to (Ahmad et al., 2020) In the current fake news corpus, there have been multiple instances where both supervised and unsupervised learning algorithms are used to classify text. However, they state that, most of the literature focuses on specific datasets or domains, most prominently the politics domain. Therefore, the algorithm trained works best on a particular type of article's domain and does not achieve optimal results when exposed to articles from other domains. Since articles from different domains have a unique textual structure, it is difficult to train a generic algorithm that works best on all particular news domains. In their paper, they propose a solution to the fake news detection problem using the machine learning ensemble approach.

Their study explores different textual properties that could be used to distinguish fake contents from real. By using those properties, they train a combination of different machine learning algorithms using various ensemble methods that are not thoroughly explored in the current literature. The ensemble learners have proven to be useful in a wide variety of applications, as the learning models have the tendency to reduce error rate by using techniques such as bagging and boosting. These techniques facilitate the training of different machine learning algorithms in an effective and efficient manner. They also conducted extensive experiments on 4 real world publicly available datasets. The results validate the improved performance of their proposed technique using the 4 commonly used performance metrics (namely, accuracy, precision, recall, and F-1 score).

According to (Hakak et al., 2021) There are numerous channels available such as social media, blogs, websites, etc., through which people can easily access the news. It is due to the availability of these platforms that the dissemination of fake news has become easier. Anyone using these platforms can create and share fake news content based on personal or professional motives. To address the issue of detecting fake news, numerous studies based on supervised and unsupervised learning methods have been proposed. However, all those studies do suffer from a certain limitation of poor accuracy. The reason for poor accuracy can be attributed due to several reasons such as the poor selection of features, inefficient tuning of parameters, imbalanced datasets, etc. In their article, they have proposed an ensemble classification model for detection of the fake news that has achieved a better accuracy compared to the state-of-the-art. The proposed model extracts important features from the fake news datasets, and the extracted features are then classified using the ensemble model consisting of three popular machine learning models namely, Decision Tree, Random Forest, and Extra Tree Classifier. They achieved a training and testing accuracy of 99.8% and 44.15% respectively on the ISOT dataset. For the Liar dataset, they achieved the training and testing accuracy of 100%.

According to (Kaliyar et al., 2019) in their research paper, firstly, they have investigated the existing models for fake news detection using content and context-based information. After an initial investigation, they have performed extensive experiments using a multi-class dataset (FNC-based fake news dataset) and employed different machine learning algorithms. In their exploration, they have found that among the different machine learning algorithms used, Gradient Boosting with optimized parameters performs the best for a multi-class fake news dataset. In the existing research, benchmark results are available based on two classes dataset, classifying news as fake or real. Work on multi-class prediction is limited. There is a huge scope of improvement for multi-class fake news detection. Their research is an attempt to improve the existing fake news classification using a multi-class dataset with the motivation that it can be helpful for future researchers working in this area.

According to (Roy et al., 2018) most of the existing studies on fake news detection are based on classical supervised model. In recent times there has been an interest towards developing deep learning based fake news detection system, but these are mostly concerned with binary classification. In their paper, they attempt to develop an ensemble-based architecture for fake news detection. The individual models are based on Convolutional Neural Network (CNN) and Bi-directional Long Short-Term Memory (LSTM). The representations obtained from these two models are fed into a Multi-layer Perceptron (MLP) for multi-class classification.

4. Legitimacy Ensemble Model

As described in (Ramkissoon & Goodridge, 2021), their research analyses the performance of an ensemble learning model for fake news detection based upon models proposed by Microsoft Azure Machine Learning Studio (classic)(AzureML). AzureML is a collaborative, drag-and-drop tool you can use to build, test, and deploy predictive analytics solutions on your data (Martens, 2020). It publishes models as web services that can easily be consumed by custom apps or business intelligence tools. AzureML is where data science, predictive analytics, cloud resources, and data meet.

The ensemble model consists of the following classification models as described in (Martens, 2020).

4.1 Two-Class Boosted Decision Tree (BDT):

The Two Class Boosted Decision Tree model has been proposed by Microsoft Azure Machine Learning Studio (classic). This paper utilises the Two Class Boosted Decision Tree based upon the results stated in (Ramkissoon & Mohammed, 2020). According to (Ramkissoon & Mohammed, 2020) From the experiments performed and the results obtained it is noted that the Two Class Boosted Decision Tree performed the best. Hence, it can be concluded that based upon our selected dataset the Two Class Boosted Decision Tree is the best method suited for detecting and predicting Credibility Based Fake News.

As described in (Martens, 2020) A boosted decision tree is an ensemble learning method in which the second tree corrects for the errors of the first tree, the third tree corrects for the errors of the first and second trees, and so forth. Predictions are based on the entire ensemble of trees together that makes the prediction. Generally, when properly configured, boosted decision trees are the easiest methods with which to get top performance on a wide variety of machine learning tasks. However, they are also one of the more memory-intensive learners, and the current implementation holds everything in memory. Therefore, a boosted decision tree model might not be able to process the very large datasets that some linear learners can handle.

Boosting can take the form of two algorithms. According to (Dev & Eden, 2019) the first method, the AdaBoost algorithm constructs an ensemble by focusing on instances that were previously misclassified. The level of focus is determined by assigning weights to the instances in the training set. The same weight is assigned to all the instances in the training set during the first iteration. With the rise in number of iterations, rise in the weights of misclassified instances occurs. On the other hand, the weights of correctly classified instances are gradually reduced. Additionally, when making a prediction using the generated ensemble, weights are also assigned to the individual base learners by considering their overall predictive performance. Ensemble construction using AdaBoost has been adapted from the work of Sugiyama 2016. Despite the focus of boosting primarily being on bias reduction, slight variance reduction can be achieved by reweighting, as is the case in AdaBoost. Variance reduction occurs because of construction of models iteratively on randomly sampled, but reweighted, training instances. The reweighting scheme controls the amount of variance reduction. With respect to classification trees, low bias and low variance can be achieved since decision trees are a low bias and high variance technique. In their work, the ensemble constructed using AdaBoost with decision stumps as weak learners, serves as a baseline for comparing prediction metrics with that of scalable GBDT systems.

The gradient-descent based formulation of boosting methods and the corresponding models are termed as gradient boosting machines (GBMs) is the second method as described by (Dev & Eden, 2019). GBMs construct base learners iteratively by reweighting observations that were misclassified. However, GBMs differ from AdaBoost in that GBMs determine the weights by operating on the negative partial derivatives of the loss function at each training observation. These partial derivatives are also called as pseudo-residuals and an ensemble is grown iteratively using these pseudo-residuals. Consequently, the feature space is partitioned grouping similar pseudo-residuals together. While GBMs can be efficient for relatively small datasets, for much larger datasets, scalable versions are needed. XGBoost, LightGBM and CatBoost are recently developed tree-based scalable versions of GBMs designed to address this requirement. They distinguish the originally formulated GBMs that use decision trees as base learners from the scalable versions by labelling them as gradient boosted decision classifiers (GBDCs). In their work, GBDCs serve as a baseline for comparison of performance metrics with that of scalable GBDT systems, namely, XGBoost, LightGBM and CatBoost. The Two Class Boosted Decision Tree implements the decision tree algorithm and boosts the tree utilising the GBM methodology.

The GBM equation can be seen in (1).

$$(\rho_t, \theta_t) = arg \min_{\rho,\theta} \sum_{i=1}^{N} -g_t(x_i) + \rho h(x_i, \theta) \qquad (1)$$

To summarize, the complete form of the gradient boosting algorithm was formulated by Friedman. The exact form of the derived algorithm with all the corresponding formulas will heavily depend on the design choices of $\psi(y, f)$ and $h(x, \theta)$.

4.2 Two Class Neural Network

According to (Martens, 2020) A neural network is a set of interconnected layers. The inputs are the first layer and are connected to an output layer by an acyclic graph comprised of weighted edges and nodes.

Between the input and output layers you can insert multiple hidden layers. Most predictive tasks can be accomplished easily with only one or a few hidden layers. However, recent research has shown that deep neural networks (DNN) with many layers can be very effective in complex tasks such as image or speech recognition. The successive layers are used to model increasing levels of semantic depth.

The relationship between inputs and outputs is learned from training the neural network on the input data. The direction of the graph proceeds from the inputs through the hidden layer and to the output layer. All nodes in a layer are connected by the weighted edges to nodes in the next layer.

To compute the output of the network for a particular input, a value is calculated at each node in the hidden layers and in the output layer. The value is set by calculating the weighted sum of the values of the nodes from the previous layer. An activation function is then applied to that weighted sum. The softmax function can be defined as:

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^{c} e^{x_j}} \qquad (2)$$

According to (Abiodun et al., 2018) The ANN is found to be a very novel and useful model applied to problem-solving and machine learning. ANN is an information manager model that is similar to biological nervous systems function of the man brain. Recently, research interest in brain functionality has rapidly increased globally. According to Haykin, an ANN can be comparable machine produced to function the same way the human brain performs a given task of interest. For example, "the human brain is big and highly efficient.

4.3 Mixture of Experts

According to (Yuksel et al., 2012) they describe the original ME regression and classification models. In the ME architecture, a set of experts and a gate cooperate with each other to solve a nonlinear supervised learning problem by dividing the input space into a nested set of regions used for classification. The gate makes a soft split of the whole input space, and the experts learn the simple parameterized surfaces in these partitions of the regions. The parameters of these surfaces in both the gate and the experts can be learned using the EM algorithm.

4.4 Logistic Regression

According to (Martens, 2020) Logistic regression is a well-known statistical technique that is used for modelling many kinds of problems. This algorithm is a supervised learning method; therefore, you must provide a dataset that already contains the outcomes to train the model.

Logistic regression is a well-known method in statistics that is used to predict the probability of an outcome and is especially popular for classification tasks. The algorithm predicts the probability of occurrence of an event by fitting data to a logistic function.

According to (Kirasich, Smith & Sadler, 2018) Linear models are composed of one or multiple independent variables that describes a relationship to a dependent response variable. Mapping qualitative or quantitative input features to a target variable that is attempted to being predicted such as financial, biological, or sociological data is known as supervised learning in machine learning terminology if the labels are known. One of the most common utilized linear statistical models for discriminant analysis is logistic regression.

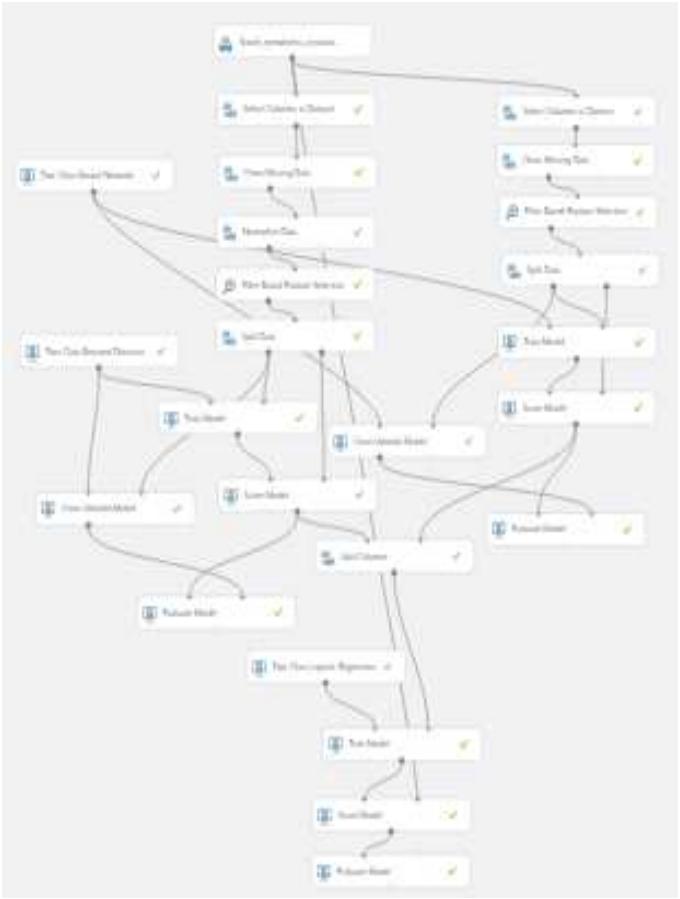

Figure 1: Experimental Setup

Simplicity and interoperability of logistic regression can occasionally lead to outperforming other sophisticated nonlinear models such as ensemble learners or support vector machines. However, in the event the response variable is drawn from a small sample size, then logistic regression models become insufficient and performs poorly for binary responses. A number of learning algorithms could be applied to modelling binary classification data types; however, the focal point of this work is to examine one linear model, logistic regression.

By combining the above individual learning models the Legitimacy ensemble model presented in this paper is built. Legitimacy combines the functionality of a Two Class Boosted Decision Tree and a Two Class Neural Network. The ensemble model is designed as seen in Fig. 1.

From the dataset the features are separated into two distinct subgroups namely demographic features and social behaviour features. The ensemble model is built based upon these two separable groups, with the aim of maximizing the performance of each individual model based upon one of the subgroups. Once the data has been separated the normal data processing rules follow from here where the data is cleaned and normalized. The data is split using a 65%-35% split for training and testing with 65% being used for training and 35% being used for testing. The Two Class Boosted Decision Tree is applied to the social behaviour data features, given its individual excellent performance with this subgroup.

The Two Class Neural Network model is applied to the demographic features as it shows its excellent behaviour with this group of features. Following the rules of machine learning and ensemble learning each individual model is trained using the subgroup of features chosen. Both models are then scored and evaluated individually. This scored dataset is scored with the testing dataset. Each scored dataset is combined to form a larger scored dataset and this amalgamated scored dataset is used to train the logistic regression model. The Logistic Regression uses the output of both the models and then trains itself with this data.

As stated in (Couronné et al., 2018) Logistic Regression can be defined using the formula found in (3).

Let Y denote the binary response variable of interest and X_1, ..., X_p the random variables considered as explaining variables, termed features in this paper. The logistic regression model links the conditional probability $P(Y = 1|X_1, \ldots, X_p)\ to\ X_1, \ldots, X_p$ through

$$P(Y = 1|X_1, \ldots, X_p) = \frac{\exp(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)}{1 + \exp(\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p)} \quad ,(3)$$

where β_0, β_0, ... , β_p are regression coefficients, which are estimated by maximum-likelihood from the considered dataset. The probability that $Y = 1$ for a new instance is then estimated by replacing the β's by their estimated counterparts and the X's by their realizations for the considered new instance in (3).

The Logistic Regression model work as the gating model in this pseudo mixture-of-experts model. The Logistic Regression uses the merged dataset and performs a weighted combination to train itself. The scored features of the merged dataset will be the list of X's as seen above. The class variable, in this case whether the news is true or fake, is the predicted Y, that this work is attempting to achieve. The Logistic Regression model chooses the percentage of each data feature to use in each situation and applies this to training the model. The model is scored and evaluated. The scored result of this Logistic Regression is the utilised as the finally predicted value of the model.

5. Ensemble Based Veracity Architecture

This research presents the Ensemble Based Veracity architecture for fake news detection in MANET messaging. The architecture accomplishes the task of gathering the credibility data of the news publisher and the content-based data of the intended message of the publisher. This is done in an infrastructureless MANET network setting in a decentralized, social computing environment utilising computational social intelligence. Veracity attempts to model social behaviour and human reactions to news spread over a MANET. Veracity utilises a prediction model along with the data features to predict whether the news is fake or not.

Simplicity and interoperability of logistic regression can occasionally lead to outperforming other sophisticated nonlinear models such as ensemble learners or support vector machines. However, in the event the response variable is drawn from a small sample size, then logistic regression models become insufficient and performs poorly for binary responses. A number of learning algorithms could be applied to modelling binary classification data types; however, the focal point of this work is to examine one linear model, logistic regression.

By combining the above individual learning models the Legitimacy ensemble model presented in this paper is built. Legitimacy combines the functionality of a Two Class Boosted Decision Tree and a Two Class Neural Network. The ensemble model is designed as seen in Fig. 1.

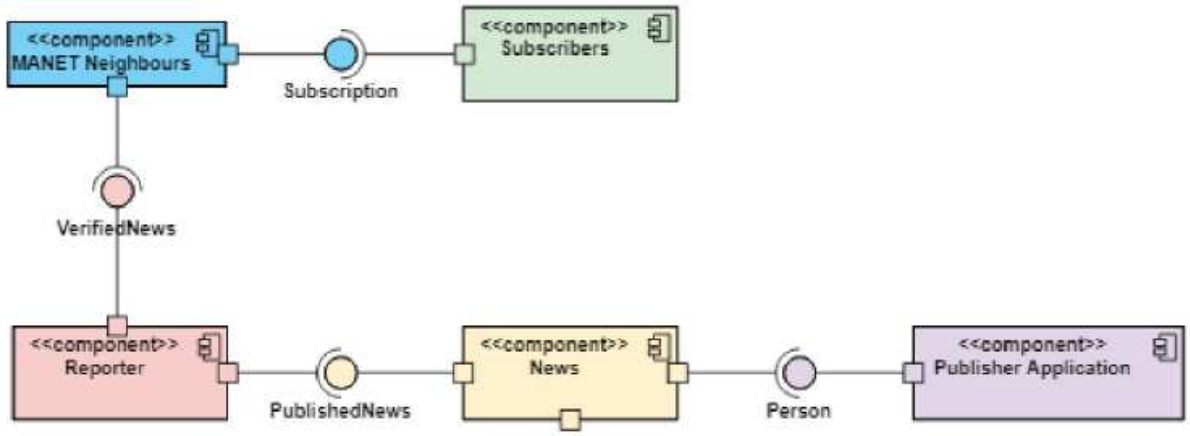

Figure 2: Components of the Veracity Architecture

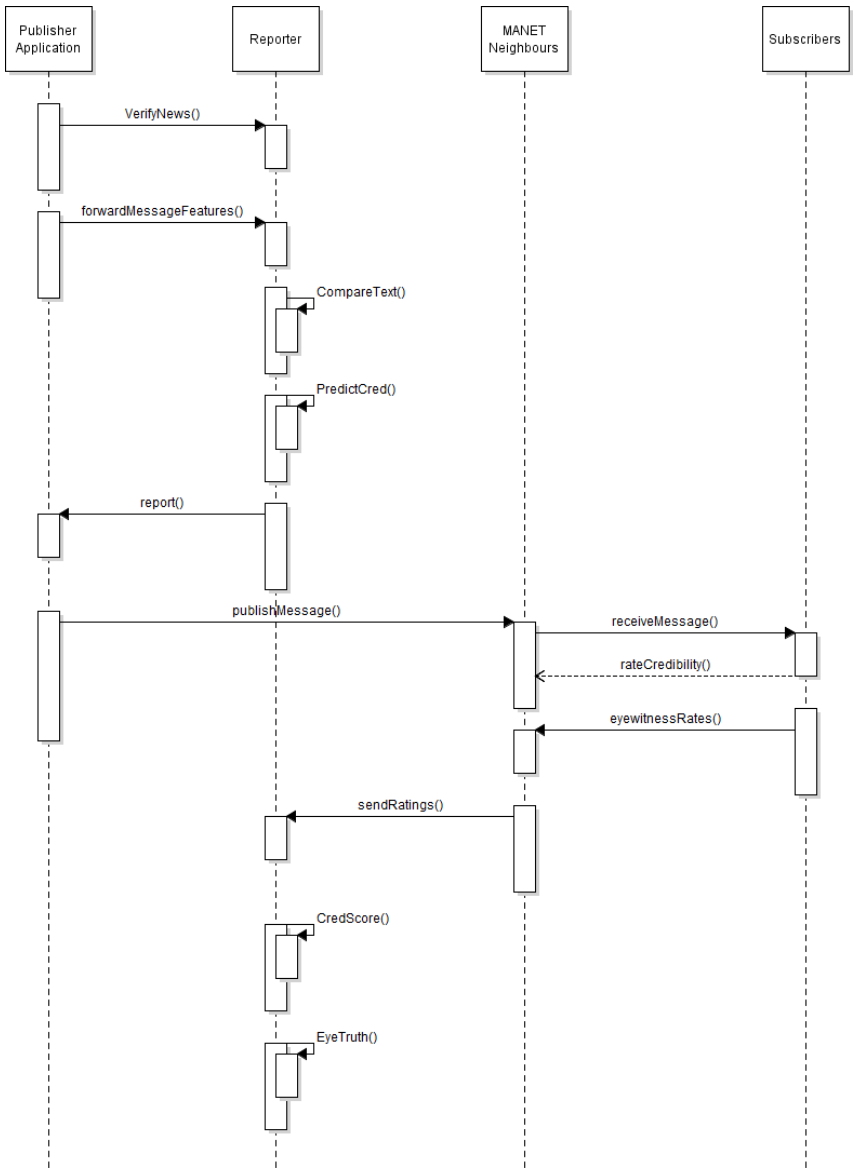

Figure 3: Interactions of the Veracity Architecture

Veracity involves the detection of fake news from the publisher. The components of Veracity can be seen in Fig. 2 above. Veracity is composed of the publisher application, the news, the monitoring agent Reporter, the various MANET neighbours, and the subscribers to the topic. On the sender's device a monitoring agent, Reporter, is downloaded. Reporter is downloaded when the publisher joins the network. The purpose of Reporter is to execute the functionality of Veracity on the sender's device. Once Reporter is installed, it will silently monitor the news the publisher intends to publish to the network. Reporter's monitoring steps can be seen in the VerifyNews algorithm below.

The monitoring is first achieved by comparing the text similarity of the message and keywords from the keyword tree and computing their similarity score. These keywords are derived based upon the topic of the message. The similarity score is computed by the CompareText algorithm as seen below.

Algorithm 1 VerifyNews Algorithm

Input: *Message M, pubfeatures [ ]*

Output: *pred*

1. For a given message M
    a. M_s = CompareText(M)
    b. *pred* = PredictCred(M_s, $pubfeatures$)
    c. If *pred* = *true*
        i. Send message M to subscribers
        ii. Wait t for ratings r_{Mi} and e_w
        iii. $P_c = CredScore(r_{Mi})$
        iv. $pubfeatures\ [i] = P_c$ // where i is the location of P_c
        v. $E_s = EyeTruth(e_w)$
        vi. $features\ [j] = E_s$ // where j is the location of E_s
2. Return *pred*

The CompareText algorithm as shown below takes the message being posted by the publisher to the network and compares this message to a list of keywords. These keywords are derived from the keyword tree based upon the topic to which the publisher is publishing the message. The message M is captured by the algorithm and converted into a vector A. The keywords T_K for the message topic M_T are also accepted by the algorithm and are converted into a vector B. Using these two vectors the cosine similarity score between these vectors is calculated to determine how similar to the intended topic the message is. The cosine similarity is found using (1)

$$M_s = A.B = ||A|||B|| \cos\theta \qquad (1)$$

The similarity score is combined with the historic credibility score, the publisher's credibility values and the message features to detect the legitimacy of the news. This detection is achieved by using the PredictCred algorithm shown below.

The dataset is firstly cleaned for missing data. The data is then normalized and then it is split. The best features are selected for use by the model and then the model is trained, scored, and evaluated. The prediction is done by a prediction model stored on the publisher's device.

If the message has been predicted as legitimate the message will be allowed to be posted to the network. Once the message has been posted to the network the users are now in control. Users will share the message, like the message, list the message etc. incrementing these counts for the said publisher for a period of time. Alongside these counts the subscribers or receivers of the messages rate the message on a scale of 1-5 stating their likeness for the message.

Algorithm 2 CompareText Algorithm

Input: M, T_K [ ]

Output: M_S

1. Retrieve from publisher T_K [ ]
2. Convert M into a vector A
3. Convert T_K into a vector B
4. $M_s = A.B = ||A|||B|| \cos\theta$
5. **return** M_S

Algorithm 3 PredictCred Algorithm

Input: M, *features* [ ]

Output: *pred*

1. Execute chosen Legitimacy ensemble learning model using M and *features*
2. **return** prediction *pred*

This rating, known as the message rating r, is returned to the publisher's Reporter. The publisher's VerifyNews algorithm periodically calls on CredScore to recalibrate the publisher's credibility score. VerifyNews waits a period of t seconds to capture a list of credibility ratings $r[\,]$. VerifyNews captures each of these ratings and passes them on for use by CredScore to calculate the Publisher Credibility Score.

According to the CredScore algorithm, the algorithm takes as input the previous mean $\bar{x}_{Mi}$ and standard deviation s_{Mi} for a given message M_i. The algorithm also accepts as input the previous number of message ratings n_{Mir} and the previous number of credibility ratings n_{cr} along with the list of message ratings $r_{Mi}[\,]$ as well as the list of message credibility scores $M_c[\,]$. The algorithm uses these ratings to calculate the mean and standard deviation of $r_{Mi}[\,]$.

The previous and new means are combined using (2).

$$\bar{X}_c = \frac{n_1\bar{X}_1 + n_2\bar{X}_2}{n_1 + n_2} \qquad (2)$$

Where $\bar{X}_c$ is the combined mean $\bar{x}_{Mil}$, $\bar{X}_1$ is the previous mean $\bar{x}_{Mi}$, n_1 is the previous number of message ratings n_{mr}, $\bar{X}_2$ is the mean of $r_{Mi}[\,]$ and n_2 is the size of $r_{Mi}[\,]$.

Once the updated mean has been calculated the updated standard deviation is also calculated using the (3).

Algorithm 4 CredScore Publisher Algorithm

Input: r_{Mi} [ ]

Output: P_c

1. Retrieve from publisher n_{Mir}, n_{cr}, M_c [ ], $\bar{x}_{Mi}$, s_{Mi}
2. For a given message M_i

    6.1 Calculate the updated mean $\bar{x}_{Mil}$ using (2) and updated standard deviation s_{Mil} using (3) and r_{Mi}.

    6.2 $M_{cMi} = \bar{x}_{Mil} + z\frac{s_{Mil}}{\sqrt{n_{Mir}}}$

    6.3 $M_c[j] = M_{cMi}$

3. $P_c = \bar{x}_p + z\frac{s_p}{\sqrt{n_{cr}}}$

4. **return** P_c

$$S_c = \sqrt[2]{\frac{n_1[{S_1}^2+(\bar{X}_1-\bar{X}_c)^2]+n_2[{S_2}^2+(\bar{X}_2-\bar{X}_c)^2]}{n_1+n_2}} \quad (3)$$

Where $\bar{X}_c$ is the combined mean $\bar{x}_{Mil}$, $\bar{X}_1$ is the previous mean, n_1 is the previous number of message ratings n_{mr}, $\bar{X}_2$ is the mean of $r_{Mi}[\,]$, n_2 is the size of $r_{Mi}[\,]$, S_1 is the previous standard deviation, S_2 is the standard deviation of $r_{Mi}[\,]$ and S_c is the combined standard deviation s_{Mil}.

Once all of these parameters have been recalculated the message credibility M_{cMi} is computed using the one-sided confidence interval formula as stated in (4).

$$M_{cMi} = \bar{x}_{Mil} + z\frac{s_{Mil}}{\sqrt{n_{Mir}}} \quad (4)$$

Where M_{cmi} is the message credibility for a given message M_i, $\bar{x}_{Mil}$ is the mean, z is the z-score value s_{Mil} is the standard deviation and n_{Mir} is the size of the combined population.

Having calculated the new message credibility score, this score is replaced with the previous one stored in $M_c[\,]$ and a new publisher credibility score P_c is calculated using to (5).

$$P_c = \bar{x}_p + z\frac{s_p}{\sqrt{n_{cr}}} \quad (5)$$

Where P_c is the message credibility, $\bar{x}_p$ is the mean, z is the z-score value s_p is the standard deviation and n_{cr} is the size of the population. The CredScore algorithm seen above returns the updated publisher credibility P_c.

Alongside the CredScore algorithm, Veracity also engages the EyeTruth algorithm. The purpose of this algorithm is to calculate the true nature of the event based upon eyewitness reports. Though Veracity provides the predicted truth, there is a need to benchmark this prediction against the actual truth. Since a subscriber can also be an eyewitness, eyewitness reports are submitted to Reporter by subscribers and hence the degree of truth is calculated using EyeTruth. The VerifyNews algorithm waits a period of t seconds to capture a list of eyewitness reports $e_w[\,]$. Each eyewitness rating in this list e is multiplied by the publisher credibility of the eyewitness P_c to give a weighted rating e_w which is forwarded to EyeTruth.

According to the EyeTruth algorithm, seen below, the algorithm takes as input the previous mean $\bar{x}_e$ and standard deviation s_e. The algorithm also accepts as input the previous number of eyewitnesses counts E_c along with the list of message ratings $e_w[\,]$. Once the algorithm has these pieces of information the mean and standard deviation of $e_w[\,]$ is calculated. The previous mean $\bar{x}_e$ and the new mean are combined to produce $\bar{x}_{el}$ using equation (2).

Where $\bar{X}_c$ is the combined mean $\bar{x}_{el}$, $\bar{X}_1$ is the previous mean $\bar{x}_e$, n_1 is the previous number of message ratings E_c, $\bar{X}_2$ is the mean of $e_w[\,]$ and n_2 is the size of $e_w[\,]$.

Once the updated mean $\bar{x}_{el}$ has been calculated the updated standard deviation s_{el} is also calculated using the (3).

Algorithm 5 EyeTruth Algorithm

Input: $e_w\,[\,]$

Output: E_s

1. Retrieve from publisher E_c, $\bar{x}_e$, s_e
2. Calculate the updated mean $\bar{x}_{el}$ using (2) and updated standard deviation s_{el} using (3).
3. $E_s = \bar{x}_{el} + z\frac{s_{el}}{\sqrt{n_e}}$
4. **return** E_s

Where $\bar{X}_c$ is the combined mean $\bar{x}_{el}$, $\bar{X}_1$ is the previous mean $\bar{x}_e$, n_1 is the previous number of ratings E_c, $\bar{X}_2$ is the mean of $e_w[\,]$, n_2 is the size of $e_w[\,]$, S_1 is the previous standard deviation s_e, S_2 is the standard deviation of $e_w[\,]$ and S_c is the combined standard deviation s_{el}.

Once all of these parameters have been recalculated the eyewitness score E_s is calculated using the one-sided confidence interval formula as stated in (6).

$$E_s = \bar{x}_{el} + z\frac{s_{el}}{\sqrt{n_e}} \qquad (6)$$

Where E_s is the message credibility, $\bar{x}_{el}$ is the mean, z is the z-score value s_{el} is the standard deviation and n_e is the size of the population.

Once this calculation is completed the updated eyewitness score E_s is returned.

Once these scores are all calculated they are updated at the publisher for use the next time the publisher publishes a message. The message flow and function calls of Veracity are illustrated in Fig. 2 which has a time complexity of $O(n)$.

6. Experimental Design

The Veracity architecture is designed as a multidimensional computational social system. This paper however only explores one dimension of the Veracity architecture, i.e., its ability to detect and predict fake news generated in MANET messaging. As such the experiment design of this work only test the ability of the architecture to achieve its main purpose.

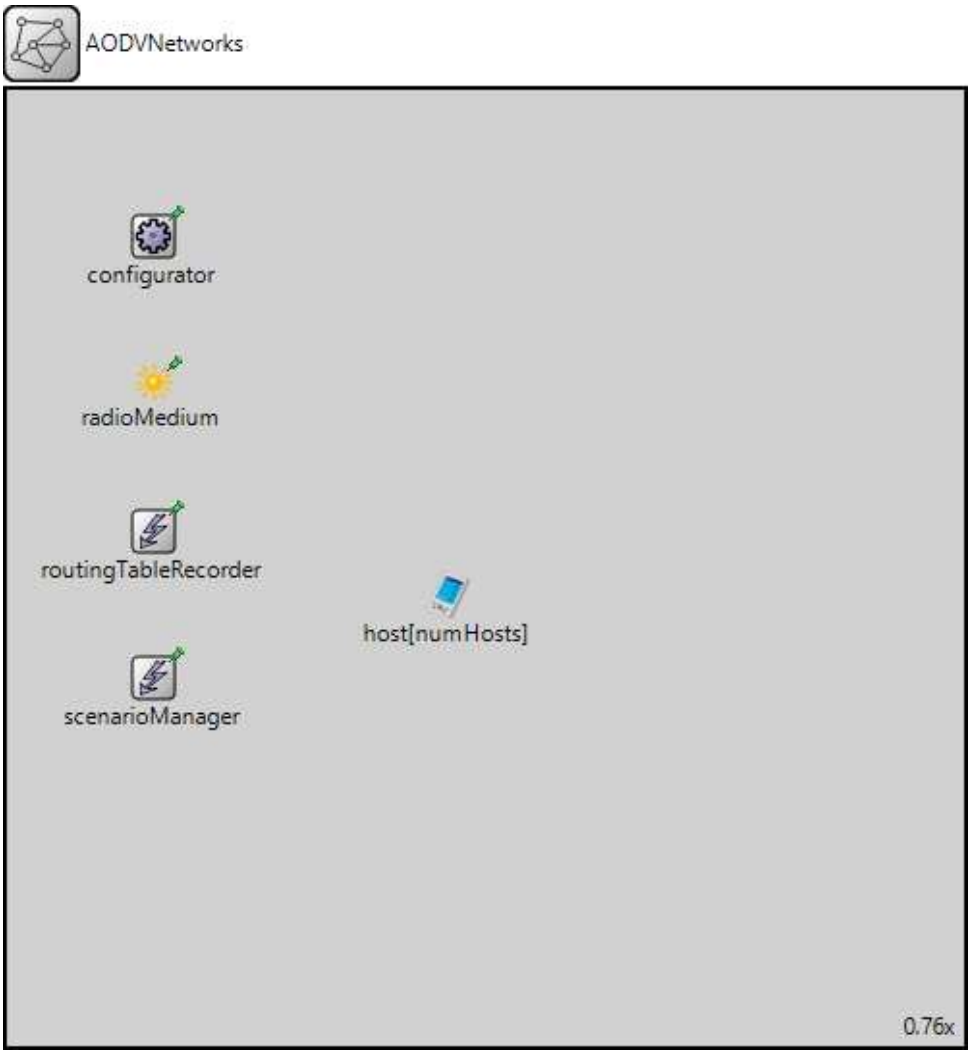

Figure 4: OMNET++ Environment.

The following are the test environment parameters:

The environment as seen in Fig. 4 comprises the use of OMNET++ for the purpose of building a MANET simulation. This simulation involves the use of AODV as a MANET routing protocol and implementing the Veracity architecture alongside it.

6.1 ML Model

For the prediction model the Microsoft Azure Machine Learning Studio (classic) environment is used to predict using the Legitimacy ensemble model as proposed by (Ramkissoon & Goodridge, 2021). The Legitimacy is chosen based upon the results presented in (Ramkissoon & Goodridge, 2021). This model is based upon Gradient Boosting Machines, Neural Networks and Logistic Regression. Gradient Boosting Machines (GBMs) as defined in (Datta and Si, 2019) are based upon the technique of gradient boosting that allows for an ensemble of trees to be developed and used for precision training. According to (Han et al, 2011)) a Neural network is a set of connected input/output units in which each connection has a weight associated with it. Logistic Regression is defined by (Martens, 2020) as a well-known statistical technique that is used for modelling many kinds of problems.

6.2 Dataset

The dataset produced mimicked that as proposed by (Zahra, Imran, and O Ostermann, 2020) and contains 17,551 records. The features generated include: the text, eyewitness, label, source, date/time, language, listed count, location, statuses count, followers count, favourites count, time zone, user language, friends count, screen name, credibility score, text similarity and eyewitness score.

6.3 Libraries

The FogNetSim++ library as proposed by (Qayyum et al., 2018) is utilised to provide the publish/subscribe functionality of the network.

The experiment is conducted using the above environmental setup. The MANET consists of 20 nodes all communicating with each other. Each node is configured as an AODV Router with the Veracity functionality added to them. The data from the dataset is cleaned for missing data. The data is then split using a separation threshold of 65% for training data and 35% for testing data. The selected ML model is then trained using the training data after which the model's performance is scored and evaluated against that of the testing dataset.

7. Results & Observations

For Veracity the results are shown below. The results are evaluated based upon the Accuracy, Precision, Recall, F1-score, and AUC values and three types of graphs.

7.1 ROC Curve

As illustrated by Fig. 5, the results are first analysed based upon the receiver operating characteristics (ROC) curve generated based upon the performance of the model. A ROC curve is an ML evaluation method that visualizes, organises, and selects classifiers based on their performance at the task of classification. ROC curves have a long history of usage in fields such as medical decision making, diagnostic systems and signal detection theory (Fawcett, 2006).

The graph below illustrates the comparison of the Sensitivity against the Selectivity for the classifier by plotting them against each other. As defined by (Gaonkar et al., 2019), Sensitivity or True Positive Rate is the ratio of true positives seen in the dataset versus the predicted number of positive values. In other words, of all the how many positive values did the ML model predict versus how many are actually contained in the dataset. Selectivity or Specificity or True Negative Rate is defined by (Gaonkar et al., 2019) as the ratio of true negatives seen in the dataset versus the predict number of negative values. In other words, how many negative values did the ML model predict versus how many are actually found in the dataset.

As stated by (J-Martens, 2020) the performance of the ML classifier can be gauged by how close the curve is to the upper left corner. If the curve is very close to the upper left corner, then the ML classifier's performance can be seen as excellent and vice versa. From the plot it can be seen that, the Sensitivity increases rapidly when the Selectivity is at zero. It then levels off and forms a plateau approximately equal to a rate of one as the False

Positive Rate increases to one. The curve formed is close to the upper left corner, illustrating excellent performance by the ML model used in the Veracity architecture.

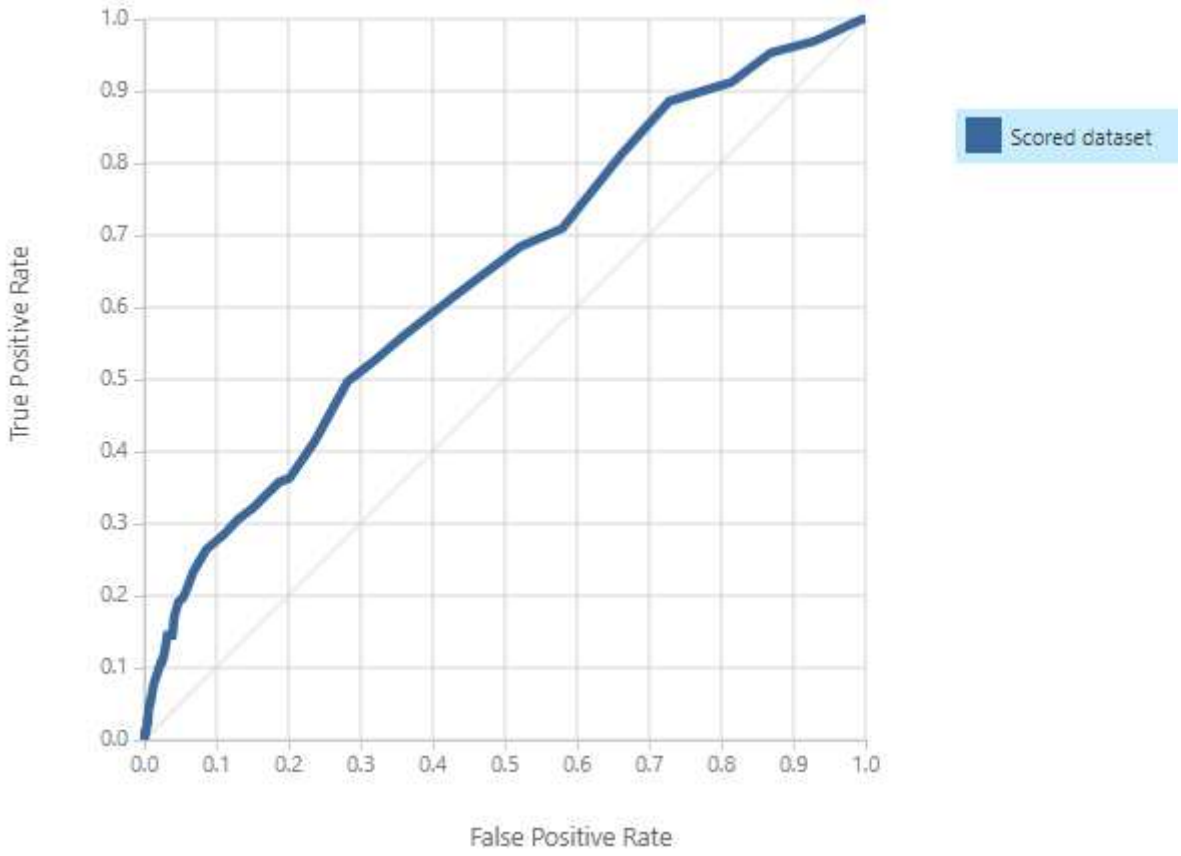

Figure 5: ROC Curve.

7.2 Precision/Recall Curve

The second method employed to analyse the performance of Veracity is the Precision/Recall Curve as seen in Fig. 6. Precision is the percentage of the quantity of true positives predicted by the ML model divided by the total number of positives predicted by the model i.e., the total value calculated when the true positive and false positive values are added together (Brownlee, 2019). It describes how good a model is at predicting the positive class. Precision is also defined as the positive predictive value. Recall is calculated as the percentage of the number of true positives predicted by the ML model divided by the total number of actual positives in the dataset i.e., the total value calculated when the true positive and false negative values are added together. Recall is the same as sensitivity.

The precision/recall curve is defined in (Brownlee, 2019) as a plot of the recall (x-axis) against the precision (y-axis) for different performance values of the classifier, similar in nature to the ROC curve. As stated by (Ekelund, 2017) Precision-recall curves often tend to produce zigzag curves that appear as erratic behaviour. This curve frequently goes up and goes down giving the zigzag impression. Therefore, when comparing multiple classifiers, precision-recall curves tend to crossover each other several times, many more times than the ROC curves making comparisons difficult. As can be seen in Figure 4 the precision-recall curve generated for the Veracity architecture is zigzagged in nature. Hence as stated above it makes it hard to compare it against the scored dataset. Therefore, a conclusive statement on the performance of Veracity from this analysis cannot be made.

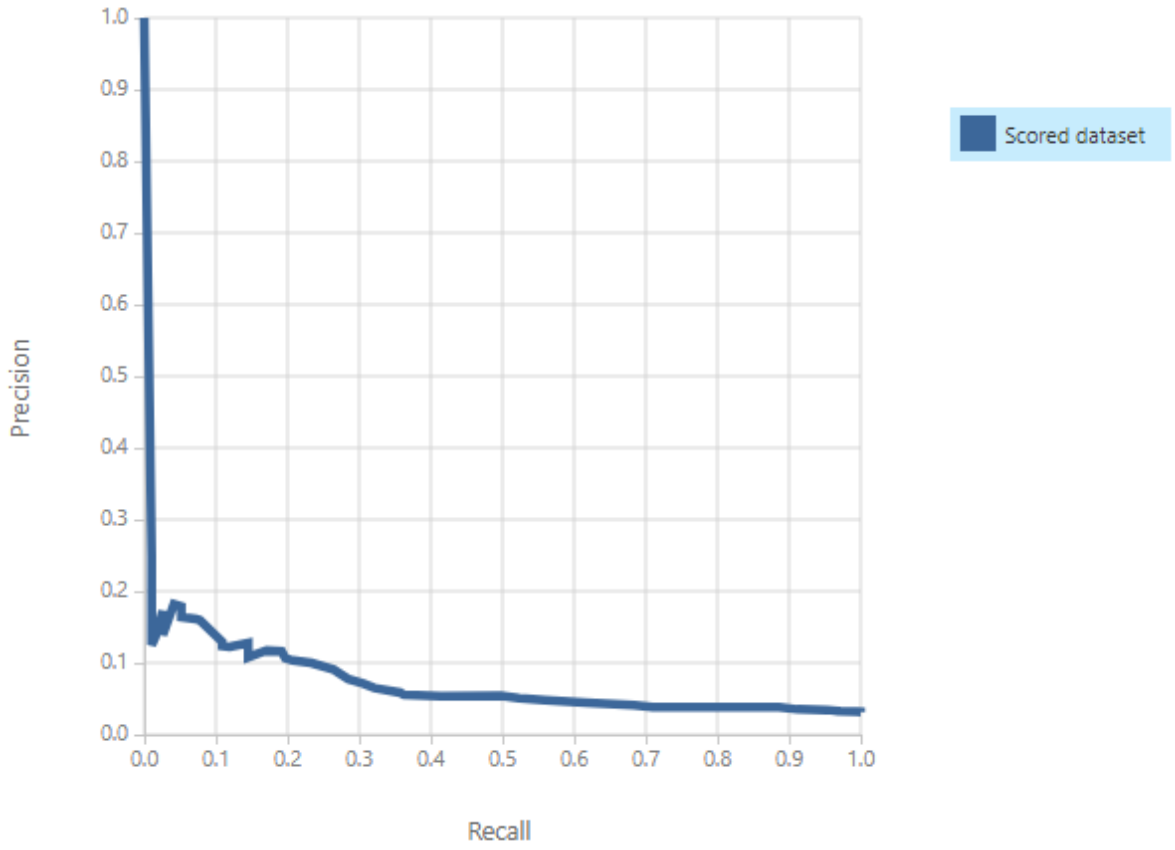

Figure 6: Precision/Recall Curve

7.3 Lift Curve

The third type of analysis is Lift Curve analysis. The Lift Curve as defined by (Choudhary and Gianey, 2017) provides a powerful performance measurement tool that measures the effectiveness of the classifier model by looking at the ratio of the target response versus the average response. It is the ratio of the result obtained with and without the classifier model applied to the prediction task known as the lift score. The graph consists of a baseline that illustrates the average response of the model, and the performance of the classifier is evaluated based upon the value of lift the model is able to produce i.e., how much better, or worst it performs with the target response.

According to (Minewiskan, 2020), a lift chart presents a graphical representation of the results obtained by a random guess versus that obtained by the use of the ML mining model. The Lift curve effectively measures the change in terms of a lift score. By comparison of the lift scores for different models, the best model can be determined. From the lift curve analysis, the points at which the model's predictions become less impactful as time goes along can also be surmised.

Just like the ROC curve, if the curve is seen to been closer to the upper left corner, then the classifier is considered good. If the curve is far away from the upper left corner, then it is not a good classifier.

The lift curve is presented in Fig. 7.

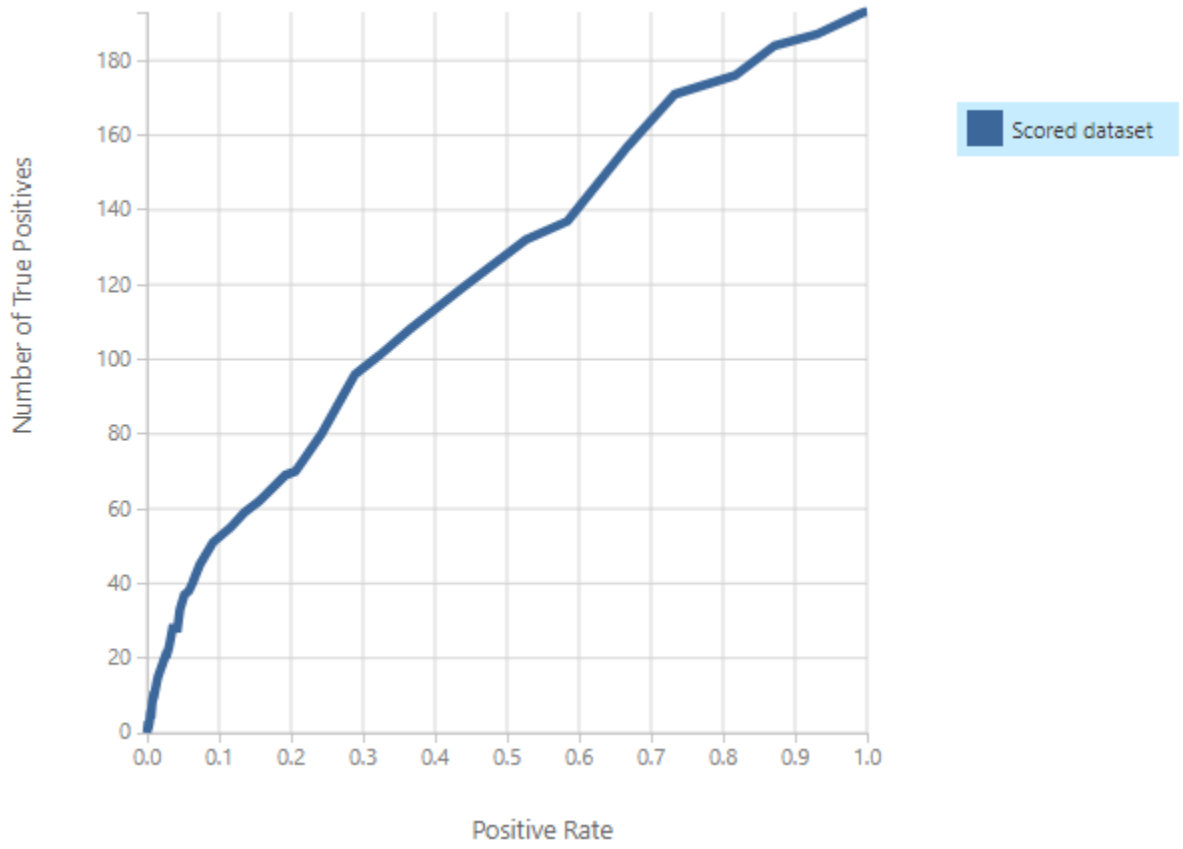

Figure 7: Lift Curve

7.4 Evaluation Metrics

The final type of results analysis used to analyse the experimental results are five evaluation metrics. For this type of analysis, the positive value is identified by 't', the negative value is identified by 'f', and the threshold value is 0.5 for all methods analysed. Accuracy or value of correctness is defined by (Vuk and Curk, 2006) as:

$$Accuracy = \frac{TP+TN}{P+N} \quad (7)$$

Where TP means the number of true positives predicted by the model, TN is the number of true negatives predicted by the classifier, P the total of positives and N the total of negatives.

Precision or value of trueness is defined by (Vuk and Curk, 2006) as:

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

Where FP stands for the false positives predicted by the model.

Recall or true positive rate is defined by (Vuk and Curk, 2006) as:

$$Recall = \frac{TP}{P} \quad (9)$$

The F1 Score or the harmonic mean of precision and recall is defined by (D. Zhang, Wang, and Zhao, 2015) as:

$$F1\ Score = \frac{2*precision*recall}{precision+recall} \quad (10)$$

The Area Under the Curve (AUC) value as defined by (Google, 2020) as a comprehensive measure of performance of the model by measuring the area of the shape formed by the curves produced in the experimental analysis. One

way of interpreting the AUC is as a measure of the likelihood that the model ranks a random negative example lower than a random positive example. The AUC value ranges in value from 0 to 1. A model whose predictions are completely wrong has an AUC of 0.0; one whose predictions are completely correct has an AUC of 1.0.

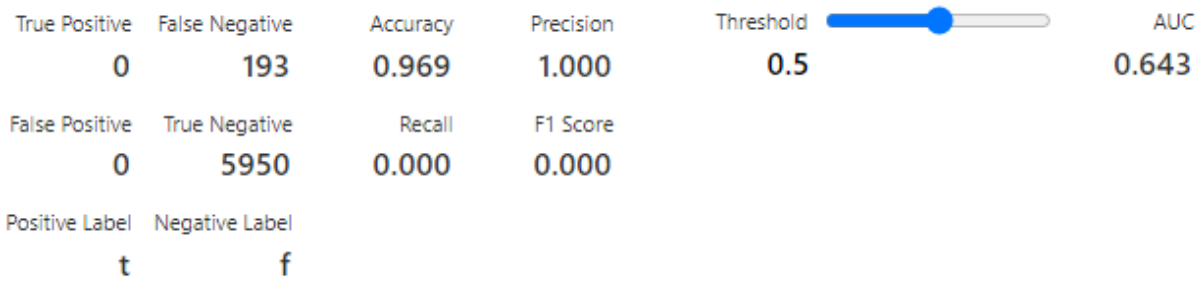

Figure 8: Evaluation Metrics

The results show that Veracity performs well at the task of detection. The accuracy of the predictive algorithm was seen to be at 96.9%. The experiment had a 100% Precision and a 0% Recall. The F1-Score was seen to be at 0% and the AUC value was seen to be at 0.643. These results indicate that the predictive model was highly accurate at the task of credibility based fake news prediction as well as being highly precise for the same task. This can also be identified by the fact that only 193 values were mis-classified and 5950 being classified correctly leading to a high accuracy value.

8. Conclusion

This research attempted to introduce an ensemble based computational social system to detect fake news in MANET messaging. As such this work introduced Veracity, a multidimensional, fake news detection architecture for MANET messaging. Veracity attempted to model social behaviour and human reactions to news spread over a MANET. Veracity worked to capture, compute, and analyse the credibility based and content-based aspects of data posted to the network for the purpose of detecting whether the news is fake or not. This architecture operated in a fully distributed and infrastructureless environment. This computational social system introduced five algorithms namely, VerifyNews, CompareText, PredictCred, CredScore and EyeTruth. These algorithms computed features that are combined and used with Legitimacy, an ensemble learning model for prediction. The prediction results were analysed using four machine learning methodologies. From the experiments conducted and the results obtained it is noted that Veracity performed excellently and successfully modelled social reactions to news on a MANET. Hence it is concluded that based upon our preliminary results that the publisher's social behaviour is directly related to the legitimacy of his content and the Veracity architecture is an appropriate method for detecting and predicting Fake News in MANET Messaging. Future work in this area involves transforming the architecture and underlying network into a Content Delivery Network with the functionality of the Veracity architecture.

References

1. (2020) URL https://developers.google.com/machine-learning/crash-course/ classification/roc-and-auc
2. Abiodun O, Isaac A, Jantan (2018)
3. Ahmad I, Yousaf M, Yousaf S, Ahmad MO (2020) Fake news detection using machine learning ensemble methods. Complexity 2020
4. Blackstock O, Blackstock U (2021)
5. Brownlee J (2019) How to Use ROC Curves and Precision-Recall Curves for Classification in Python. Machine Learning Mastery
6. Choudhary R, Gianey HK (2017) Comprehensive review on supervised ma- chine learning algorithms. 2017 International Conference on Machine Learning and Data Science (MLDS) pp 37–43
7. Collins B, Hoang DT, Nguyen NT, Hwang D (2020) Fake News Types and Detection Models on Social Media A State-of-the-Art Survey. In: Asian
8. Conference on Intelligent Information and Database Systems, Springer, pp 562–573
9. Couronné R, Probst P, Boulesteix AL (2018) Random forest versus logistic regression: a large-scale benchmark experiment. BMC bioinformatics 19(1):1– 14

10. Datta A, Si S (2019) A Supervised Machine Learning Approach to Fake News Identification. International Conference on Intelligent Data Communication Technologies and Internet of Things pp 197–204
11. Dev VA, Eden MR (2019) Formation lithology classification using scalable gradient boosted decision trees. Computers & Chemical Engineering 128:392– 404
12. Ekelund S (2017)
13. Fawcett T (2006) An introduction to ROC analysis". Pattern recognition letters 27(8):861–874
14. Gaonkar S (2019) Detection Of Online Fake News: A Survey. 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN) pp 1–6
15. Hakak S, Alazab M, Khan S, Gadekallu TR, Maddikunta PKR, Khan WZ (2021) An ensemble machine learning approach through effective feature extraction to classify fake news. Future Generation Computer Systems 117:47–58
16. Herzig A, Lorini E, Pearce D (2019)
17. Kaliyar R, Kumar A, Goswami P, Narang (2019) Multiclass fake news detection using ensemble machine learning. 2019 IEEE 9th International Confer- ence on Advanced Computing (IACC) pp 103–107
18. Khan JM, Younus (2019) URL https://docs.microsoft.com/en-us/azure/ machine-learning/studio
19. Khan Y, Junaed T, Khondaker AI, Iqbal S, Afroz (2019)
20. Kirasich K, Smith T, Sadler B (2018) Random forest vs logistic regression: binary classification for heterogeneous datasets. SMU Data Science Review 1(3):9–9
21. Lazer D (2018) The science of fake news". Science 359(6380):1094–1096 Liang X (2020) "Introduction". Social Computing with Artificial Intelligence pp 1–7
22. Liang, Xun (2020a). "Introduction". In: Social Computing with Artificial Intelligence. Springer, pp. 1–7.
    - (2020b) Social Computing Application in Online Crowd Behavior and Psychology. Social Computing with Artificial Intelligence pp 257–276
    - (2020c) Social Computing Application in Public Security and Emergency Management. Social Computing with Artificial Intelligence pp 233–242
23. Minewiskan (2020) URL https://docs.microsoft.com/en-us/analysis-services/ data-mining/lift-chart-analysis-services-data-mining?view=asallproducts- allversions
24. Murugan R, Shanmugam (2012) Cluster based node misbehaviour detection, isolation and authentication using threshold cryptography in mobile Ad hoc networks". International Journal of Computer Science and Security (IJCSS) 6(3):188–188
25. Nazir M (2016) A novel review on security and routing protocols in MANET. Communications and Network 8(4):205–218
26. Pian W, Chi J, Ma F (2021)
27. Qayyum T (2018) FogNetSim++: A toolkit for modeling and simulation of distributed fog environment. IEEE Access 6:63570–63583
28. Ramkissoon A, Neil S, Mohammed (2020) An Experimental Evaluation of Data Classification Models for Credibility Based Fake News Detection. 2020 International Conference on Data Mining Workshops (ICDMW) pp 93–100
29. Roy A, Basak K, Ekbal A, Bhattacharyya P (2018)
30. Shu K (2017) Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter 19(1):22–36
31. Shu K, Wang S, Liu H (2018) Understanding user profiles on social media for fake news detection. 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR) pp 430–435
32. Sohail M (2019) Multi-hop interpersonal trust assessment in vehicular ad- hoc networks using three-valued subjective logic. IET Information Security 13(3):223–230
33. Stieglitz S, Fuchß C (2011) Challenges of MANET for mobile social networks. Procedia Computer Science 5:820–825
34. University R, Aachen (2021) Computational Social Systems M.Sc." Computational Social Systems. URL https://www.rwthaachen.de/go/id/sthd?lidx= 1#aaaaaaaaaaasthe
35. Vanesa Daza (2008) Trustworthy privacy-preserving car-generated announcements in vehicular ad hoc networks. IEEE Transactions on Vehicular Technology 58(4):1876–1886
36. Vuk M, Curk T (2006) "ROC curve, lift chart and calibration plot". Metodoloski zvezki 3(1):89–89
37. Xiao Y, Liu Y, Li T (2020) Edge computing and blockchain for quick fake news detection in IoV. Sensors 20(16):4360–4360

38. Yuksel S, Esen JN, Wilson PD, Gader (2012) Twenty years of mixture of experts. IEEE transactions on neural networks and learning systems 23:1177–1193
39. Zahra K, Imran M, Ostermann FO (2020) Automatic identification of eyewitness messages on twitter during disasters. Information processing & management 57(1):102107–102107
40. Zhang D, Wang J, Zhao X (2015) Estimating the uncertainty of average F1 scores. Proceedings of the 2015 International Conference on The Theory of Information Retrieval pp 317–320
41. Zhang X, Ghorbani AA (2020) An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management 57(2):102025–102025
42. Zhou X, Zafarani R (2018)