UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia "Galileo Galilei"

Master Degree in Physics of Data

Final Dissertation

Machine learning methods to predict the formation of binary compact objects

Thesis supervisor

Prof. Michela Mapelli

Thesis co-supervisor

Dr. Giuliano Iorio

Candidate

Vivek Kashyap Janardhana

Academic Year 2022/2023

Machine learning methods to predict the formation of binary compact objects

Vivek Kashyap Janardhana

Abstract

In this thesis I tested different machine learning methods (Random forest, XGBoost) to predict the formation of binary compact objects solely based on initial condition of stellar binaries (masses, metallicity and orbital parameters) and on the parameters of binary evolution (i.e., common envelope efficiency). I trained the machine learning methods on simulations performed with rapid binary population synthesis code (BPS) code SEVN. I found that the tested methods can predict the formation of merging (within the Hubble time) and non-merging binary compact objects. The performance of predicting merging compact objects is found to be 50%, which is equivalent to random chance. This suggests that the methods cannot accurately predict the formation of merging compact objects. However, the methods show good performance in distinguishing between systems that produce bound and non-bound systems.

Contents

1 Introduction **1**
1.1 Gravitational waves & GW detectors 1
1.2 Sources of GW: compact objects 3
1.3 Binaries of stellar blackhole 4
1.4 Integrating theory with SEVN to explore the parameter phase and taking assistance of ML 8

2 Machine learning **9**
2.0.1 Types of ML algorithms 9
2.0.2 Supervised and unsupervised learning 10
2.0.3 Random forest 11

3 SEVN **15**
3.1 SEVN 15
3.1.1 Binary population synthesis code 15
3.1.2 SEVN 15
3.2 Data, training and validation 20
3.2.1 Training the model 20
3.2.2 Challenges encountered while training with SEVN data 22

Chapter 1

Introduction

1.1 Gravitational waves & GW detectors

The concept of gravitational waves traces its origins back to the revolutionary theory of general relativity proposed by Albert Einstein in 1915. In 1918, Einstein made a remarkable prediction: the existence of gravitational waves as a consequence of his theory. These waves, he proposed, are ripples in the fabric of spacetime itself, propagating through the universe at the speed of light.

Einstein's prediction of gravitational waves was a profound revelation that not only challenged our understanding of gravity but also presented a new perspective on the nature of the cosmos. It indicated that gravity, far from being merely a force acting at a distance, is an intrinsic property of the spacetime continuum. Gravitational waves became an essential aspect of Einstein's general theory of relativity, expanding our understanding of the gravitational interaction and its influence on the behavior of massive objects in the universe.

The significance of gravitational waves lies in their potential to unveil the secrets of the most violent and energetic events in the cosmos. These include the collision of massive black holes, the coalescence of neutron stars, and the explosive cataclysms of supernovae. By detecting and studying gravitational waves, scientists can gain unique insights into the dynamics, properties, and evolution of these celestial phenomena.
Moreover, the detection of gravitational waves provides a powerful tool to test the validity of general relativity itself. By comparing the observed properties of gravitational waves with the predictions of the theory, researchers can verify its accuracy in extreme gravitational environments. This opens the door to exploring alternative theories of gravity and deepening our understanding of the fundamental forces that shape the universe.

When massive objects, such as binary systems composed of black holes or neutron stars, are in motion, their accelerated masses generate changes in the curvature of spacetime. This disturbance propagates outward as gravitational waves, carrying energy and information about their sources.

Gravitational waves can be understood as ripples in the fabric of spacetime itself. Just as a stone thrown into a pond creates waves that spread across the surface, the motion of massive objects disrupts the smoothness of spacetime, creating undulations that radiate away from the source.

The emission of gravitational waves occurs when the motion of massive objects involves changes in their quadrupole moment. Binary systems, in particular, are excellent sources of gravitational waves. As black holes or neutron stars orbit each other, they undergo strong gravitational interactions, causing them to spiral inward over time.

Crucially, gravitational waves carry energy away from their sources. As they propagate through spacetime, they transport energy across vast cosmic distances. This energy loss affects the orbital dynamics of binary systems, causing them to gradually lose energy and spiral closer together.

Furthermore, gravitational waves provide valuable information about their sources. The precise characteristics of the emitted waves encode details about the masses, velocities, and orbital parameters of the objects involved in the binary system. By studying the properties of gravitational waves, scientists can gain insights into the nature of these massive objects, their dynamics, and the environments in which they reside.

The first detection of gravitational waves was made by the Laser Interferometer Gravitational-Wave Observatory (LIGO) in 2015. LIGO observed the merger of two black holes, an event that occurred 1.3 billion light-years away, from [1]. This monumental achievement not only validated Albert Einstein's prediction of gravitational waves but also demonstrated the remarkable precision of LIGO's instruments in measuring tiny ripples in spacetime.

"Over the last five years, LIGO and Virgo [3] witnessed a rapidly growing number of GW events: the second gravitational wave transient catalogue (GWTC-2, [[2]]) consists of 50 binary compact object mergers from the first (O1), the second (O2) and the first part of the third observing run (O3a) of the LIGO–Virgo collaboration (LVC)." as mentioned in [8]

Another significant milestone in gravitational wave astronomy was the detection of a binary neutron star merger in 2017. This event, called GW170817, was not only observed through gravitational waves but also through a wide range of electromagnetic signals across the spectrum, including gamma rays, X-rays, and radio waves. This multi-messenger observation marked a major breakthrough, as it confirmed that gravitational waves and light travel at the same speed and provided valuable insights into the production of heavy elements, such as gold and platinum, in such collisions.

The scientific achievements resulting from gravitational wave observations have been numerous. These observations have allowed scientists to test the predictions of Einstein's general theory of relativity in extreme gravitational environments and have provided evidence for the existence of black holes and neutron stars. Gravitational waves have also served as a new tool for studying the properties of matter under extreme conditions, probing the nature of dense nuclear matter and the behavior of spacetime itself.

Furthermore, gravitational wave astronomy has fostered international collaborations and the development of advanced technologies. The global network of gravitational wave detectors, including LIGO, Virgo, and KAGRA, working together has significantly improved the sensitivity and detection capabilities, enabling more frequent observations and expanding the reach of gravitational wave science.

The impact of gravitational wave observations cannot be overstated. These groundbreaking discoveries have confirmed the existence of gravitational waves, provided unique insights into astrophysical processes, and opened up new avenues for studying the universe. From the first direct detection in 2015 to subsequent observations of binary black hole mergers and neutron star collisions, gravitational wave astronomy continues to push the boundaries of our knowledge, revolutionizing our understanding of the cosmos and inspiring new scientific endeavors.

1.2 Sources of GW: compact objects

Gravitational waves are produced by a variety of astrophysical events and phenomena, each characterized by their own unique properties and significance. The study of gravitational waves allows us to probe the dynamics and behavior of these systems, shedding light on the violent and energetic nature of the universe. Few of the possible sources of GW studied and observed are as follow:

Binary system

Binary systems composed of massive objects, such as black holes or neutron stars, are important sources of gravitational waves. These systems consist of two massive objects orbiting each other due to their mutual gravitational attraction.

The motion and orbital dynamics of the binary system create disturbances in the fabric of spacetime, causing it to ripple and propagate as gravitational waves. According to Einstein's general theory of relativity, the accelerated masses in the binary system generate changes in the curvature of spacetime, analogous to ripples spreading on the surface of a pond when a stone is thrown into it.

As the binary objects orbit each other, they continuously emit gravitational waves. These waves carry away energy from the system, causing the binary objects to gradually lose orbital energy. As a result, the separation between the objects decreases over time, and their orbits decay. This energy loss process ultimately leads to the merger of the binary system.

The emission of gravitational waves is a consequence of the binary objects converting their orbital energy into gravitational wave energy. These waves carry away the energy in the form of ripples in spacetime, propagating outward in all directions from the source. The emission of gravitational waves is a fundamental process that acts as a feedback mechanism in the binary system, regulating its dynamics and causing the objects to inspiral toward each other.

As the binary objects approach each other and their orbital velocities increase, the emission of gravitational waves becomes more intense. The emitted waves carry information about the masses, spins, and orbital parameters of the objects. By analyzing the properties of the gravitational wave signals detected on Earth, scientists can infer valuable information about the binary system, such as the masses and spins of the objects, their orbital characteristics, and the nature of the merger process.

1.3 Binaries of stellar blackhole

The evolution of binary stars does not differ much from the single stellar evolution unless either of the stars comes in the way of the other, that is if the binary orbit is large enough. Unlike in the case where the stars are close enough when they interact with each other, consequence of which is an evolution and appearance of stars along with the a change in the orbit itself. One can think that if two massive stars are members of a binary system, they will become a binary blackhole (BBH) system. In case the binaries are 'detached', essentially far from each other then the mass of each of the blackhole (BH) are same as that of the progenitor star as if they were both single stars. But if the binaries are tight, that is close enough to each other, then they can evolve in one of several possible ways. The binary population synthesis (BPS) codes is used to investigate the effects of binary evolution process on the formation of the BBH in isolated binaries. BPS codes are semi analytical codes with the prescription for supernovae explosions and formalism for binary evolution process. Few of the most important binary evolution process are:

Mass transfer

If the two stars exchange matter with each other, then it is termed as mass transfer. It is driven by stellar winds or Roche lobe filling. When massive star losses mass by stellar wind, its companion might capture some of this mass depending on amount of mass lost

- Relative velocity of the stellar wind with respect to companion star. The mean mass accretion rate by the stellar wind is given by

$$\dot{m}2 = \frac{1}{\sqrt{1-e^2}} \left(\frac{Gm_2}{v_m^2}\right)^2 \frac{\alpha_w}{2a^2} \frac{1}{\left(1+\left(\frac{v\text{orb}}{v_w}\right)^2\right)^{\frac{3}{2}}} |\dot{m}_1| \tag{1.1}$$

  Where the "$\dot{m}_2$" is the rate of mass gained by the accretion star, "G" is the gravitational constant, "e" is the eccentricity of the binary orbit,"v_m^2" velocity of the stellar wind, "m_2" is the mass of the accretion star, "α_w" is the efficiency constant taken approximately as 1.5, "a" is semimajor axis of the binary, "v_orb" is the orbital velocity given by "$v_orb = \sqrt{(G(m_1+m_2))/a}$" and "$\dot{m}_1$" donor star mass loss rate.

- Mass transfer by Roche lobe is a more efficient method than wind accretion. Roche lobe of a star in a binary system, which can be described as a teardrop-shaped equipotential surface enveloping the star. The Roche lobes of both binary members are interconnected at a single point known as the Lagrangian L1 point. To approximate the Roche lobe, a commonly utilized formula is employed:

$$R_{L,1} = a\frac{0.49q^{2/3}}{0.6q^{2/3} + ln(1+q^{1/3})} \tag{1.2}$$

  Where the parameter "a" represents the semi-major axis of the binary system, while "q" denotes the mass ratio between the two stars, given by $q = m_1/m_2$ (where m_1 and m_2 refer to the masses of the two stars in the binary). This equation characterizes

the Roche lobe of a star with mass m_1, while the corresponding Roche lobe of a star with mass $m_2(R_{L,2})$ can be obtained by interchanging the subscripts. When a star's radius exceeds the size of its Roche lobe, it is said to be overflowing the lobe, whereas if its radius is smaller, it is considered to be underfilling the lobe. In cases where a star overflows its Roche lobe, a portion of its mass is transferred towards the companion star, which may then accrete some or all of the incoming mass.
Mass transfer in a binary system has a significant impact not only on the masses of the two stars but also on the final mass of the compact remnants and the orbital characteristics of the binary. In cases where mass transfer is non-conservative, which is considered to be the more realistic scenario, it results in a loss of angular momentum, consequently affecting the semi-major axis of the binary. Recent studies **??** have revealed that the assumption of highly non-conservative mass transfer (with a mass accretion efficiency $f_{MT} \leq 0.5$) is in conflict with the data from the Laser Interferometer Gravitational-Wave Observatory (LVC), particularly when assuming that all observed BBH systems originate from isolated binary evolution.
An essential aspect to consider regarding Roche lobe overflow is determining whether it is stable or unstable and the timescale involved. To explore this further, we can make an assumption that the stellar radius and mass are linked through the relationship $R \propto m^{\zeta}$. Hence the variation in donor's radius during Roche lobe is

$$\frac{\mathrm{d}R_1}{\mathrm{d}t} = \frac{\partial R_1}{\partial t} + \zeta \frac{R_1}{m_1} \frac{\mathrm{d}m_1}{\mathrm{d}t} \tag{1.3}$$

Where the term "$\frac{\partial R_1}{\partial t}$" is the contributing term for presence of nuclear burning, while the term involving "ζ" quantifies the adiabatic or thermal reaction of the donor star to the loss of mass. It is important to note that "$\frac{\mathrm{d}m_1}{\mathrm{d}t}$" represents the rate of mass loss from the donor and is always negative. In a similar manner, we can estimate the alteration in the size of the donor's Roche lobe $R_{L,1}$ as follows:

$$\frac{\mathrm{d}R_{L,1}}{\mathrm{d}t} = \frac{\partial R_{L,1}}{\partial t} + \zeta_L \frac{R_{L,1}}{m_1} \frac{\mathrm{d}m_1}{\mathrm{d}t} \tag{1.4}$$

where $\frac{\partial R_{L,1}}{\partial t}$ depends on the tides and GW radiation, while ζ_L is the response of the Roche lobe to mass loss, as in Roche lobe might expand or shrink. If the $\zeta_L > \zeta$ then the Roche lobe shrinks faster the radius of the star also making the mass transfer unstable, else it remains stable until the radius changes dramatically by nuclear burning, Mass transfer might be unstable on a dynamical timescale, in this case ζ describes (if ζ represents an adiabatic response of the donor and $\zeta < \zeta_L$) or might be stable on a thermal timescale (one in which ζ represents the thermal response of the donor and $\zeta < \zeta_L$). In the event of dynamically unstable mass transfer or if both stars exceed their respective Roche lobes, it is anticipated that the binary system will undergo either a merger, especially when the donor lacks a significant density gradient between its core and envelope, or enter a common envelope (CE) phase, particularly when the donor exhibits a distinct separation between its core and envelope.

Common envelope

When a binary system enters the CE phase, the envelopes of both stars cease to co-rotate with their cores. Consequently, the stellar cores (or the compact object and the

core of the companion star in the case of an already single degenerate binary) become embedded within a shared non-corotating envelope, initiating a spiraling motion inward due to gas drag exerted by the envelope. As the cores experience energy loss during this process, a portion of the lost orbital energy is likely converted into heating the envelope, causing it to become less tightly bound. If this mechanism leads to the ejection of the envelope, the binary survives, but it becomes composed of two exposed stellar cores (or a compact object and a naked stellar core). Moreover, the orbital separation between the two cores (or between the compact object and the stellar core) becomes significantly smaller than the initial separation prior to the CE phase due to the inward spiral. This outcome holds significant implications for the fate of a BBH system. In fact, if a BBH survives the CE phase, its semi-major axis (a) becomes notably shorter ($a \lesssim 100 R_{\odot}$) compared to the sum of the maximum radii of the progenitor stars. Consequently, the BBH may have the potential to merge within a Hubble time through the emission of gravitational waves.
On the other hand, if the envelope is not ejected, the two cores (or the compact object and the core) continue spiraling inward until they eventually merge. This premature merger during the CE phase prevents the binary from evolving into a BBH. The schematic diagram in ?? provides an overview of these potential outcomes.

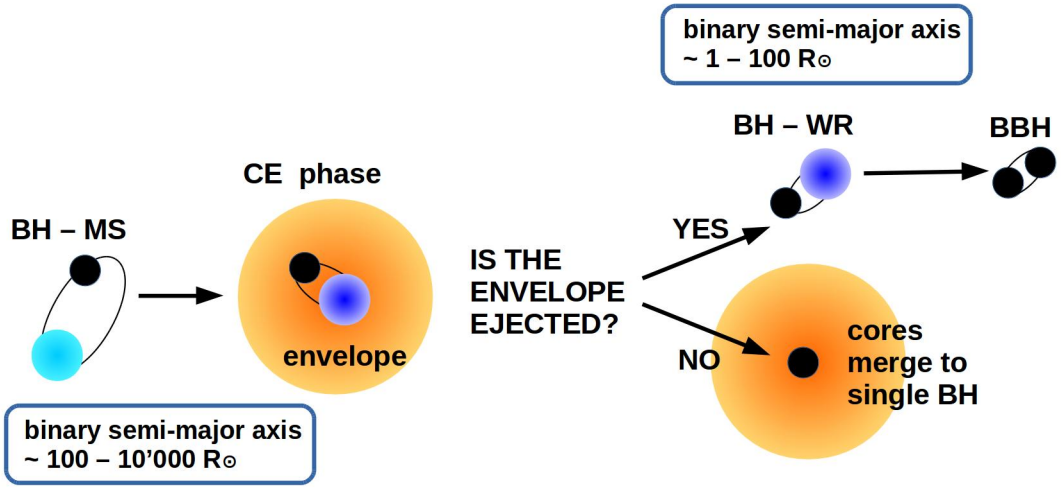


Figure 1.1: Schematic representation of the evolution of a BBH through CE. The companion of the BH is initially in the main sequence. Figure referred from [10]. Also mentioned WR is a Wolf-Rayet star(ie, almost completely stripped by the hydrogen envelope)

The most commonly adopted formalism to describe the common envelope is the α formalism. This formalism is based on the notion that the energy required to unbind the envelope originates solely from the loss of orbital energy of the two cores during the inward spiral. The fraction of the orbital energy contributed by the two cores that goes into unbinding the envelope can be expressed using the following formalism:

$$\delta E = \alpha(E_{b,f} - E_{b,i}) = \alpha \frac{G m_{c1} m_{c2}}{2} \left(\frac{1}{\alpha_f} - \frac{1}{\alpha_i}\right) \tag{1.5}$$

where $E_{b,i}(E_{b,f})$ represents the orbital binding energy of the 2 cores either before (or after) the CE phase, m_{c1} and m_{c2} are masses of the two cores, α represents the fraction of orbital energy which is transferred to the envelope and is a dimensionless parameter. As on the

??, if the primary(m_{c1}) is already a compact then it implies that the m_{c2} gives the mass of the secondary compact object.
Binding energy of the envelope is:

$$E_{env} = \frac{G}{\lambda}\left[\frac{m_{env,1}m_1}{R_1} + \frac{m_{env,2}m_2}{R_2}\right] \tag{1.6}$$

here m_1 and m_2 are the are the masses of the primary and the secondary member of the binary, $m_{env,1}$ and $m_{env,1}$, represent the masses . R_1 and R_2 represent the radii of the primary and secondary in the envelope of binary and λ is parameter representing the concentration of the envelope, lower the value of λ higher is the concentration of the envelope.

By equating δE (orbital energy change) to E_{env} (binding energy of the envelope), we can determine the value of the final semi-major axis (a_f) at which the envelope is ejected. Consequently, a larger (smaller) α initial semi-major axis (a) leads to a correspondingly larger (smaller) final orbital separation. If the resulting a_f is less than the sum of the radii of the two cores (or less than the sum of the Roche lobe radii of the cores), the binary system will undergo a merger during the common envelope phase. Conversely, if a_f is greater than or equal to the sum of the core radii (or the sum of the Roche lobe radii), the binary system will survive.

1. The thermal energy of the envelope, encompassing the combined contributions of radiation energy and the kinetic energy of gas particles
2. Recombination energy, which arises as the expanding envelope cools down, leading to plasma recombination and the release of binding energy through the formation of atoms and even molecules
3. Tidal heating/cooling resulting from stellar spin down/up
4. Nuclear fusion energy
5. Enthalpy of the envelope
6. Accretion energy, which may drive the occurrence of outflows and jets.

When the two cores undergo a rapid dynamical spiral-in process, typically occurring within a timescale of around 100 days, only a small fraction of the envelope (approximately 25% in most simulations) appears to be ejected. As the two cores approach each other closely and the gas mass between them becomes minimal, the spiral-in process slows down, and the system evolves on a much longer Kelvin-Helmholtz timescale of the envelope, spanning $\approx 10^{3-5}$ years.

Simulating the system for the entire Kelvin-Helmholtz timescale is currently infeasible for three-dimensional simulations due to computational limitations . Fragos et al. (2019,) address this challenge by employing a simplified approach, conducting one-dimensional simulations of the entire common envelope evolution using the hydrodynamic stellar evolution code MESA [270, 271, 272]. They model a binary system consisting of a neutron star (NS) and a red supergiant star with a mass of 12 M, simulating the thermal timescale of the envelope. In their model, the ejection of the envelope is primarily driven by the thermal energy of the envelope and the orbital energy, while recombination contributes to less than 10% of the total energy required for envelope ejection. The resulting system is a NS-naked helium star system with a relatively close orbital separation of a few stellar radii, making it a promising progenitor for a merging binary neutron star (BNS) system.

Their simulated system aligns with the formalism when the parameter a is set to 5, despite this value seemingly exceeding physical limits. This choice stems from the recognition that orbital energy is just one of the contributing energy sources involved in the process of envelope ejection.

1.4 Integrating theory with SEVN to explore the parameter phase and taking assistance of ML

By integrating theory with the SEVN (Stellar Evolution with N-body dynamics) approach, we aim to explore the parameter phase and leverage the assistance of machine learning (ML) techniques. This integration allows us to combine theoretical knowledge and computational simulations to gain a deeper understanding of the underlying mechanisms and relationships within the system. Through the application of ML algorithms, we can analyze the complex data generated by SEVN and extract meaningful insights, enabling us to make more accurate predictions and uncover novel patterns within the parameter space. This interdisciplinary approach provides a powerful framework for advancing our understanding of astrophysical phenomena and enhancing our ability to explore and interpret the intricate dynamics of celestial objects.

Chapter 2

Machine learning

Machine learning is the development of algorithms and models to enable a computer to learn about data and make predictions without explicitly programming it. ML enables the computer to recognize patterns in the data, fit equations to variables, categorize data and do more operations on the data it learns from. Machine learning has been used in a wide range of applications in physics from particle physics experiments such as LHC to astrophysics experiments in galaxy classification, gravitational wave detection, cosmological parameter estimation, from quantum error correction, and quantum control in quantum physics to solving complex equations, simulating physical systems, and optimizing computational models in computational physics. In traditional programming, explicit instructions are written to perform specific tasks, whereas in ML the focus is on developing models that can generalize from examples, by analyzing from the data set the ML can extract meaningful insights and recognize complex patterns and make informed predictions.

Here, we delve into the fundamental concepts of machine learning, with a particular focus on supervised learning, such as the random forest algorithm.

2.0.1 Types of ML algorithms

There are several ways of classifying machine learning algorithms, based on various criteria, a few of these ways are as follows:

- **Supervised vs Unsupervised learning vs Reinforcement learning:**
  Supervised Learning: Algorithms learn from labeled training data, where inputs are mapped to known outputs.
  Unsupervised Learning: Algorithms learn from unlabeled data, finding patterns, structures, or relationships in the data.
  Reinforcement Learning: Algorithms learn by interacting with an environment, receiving feedback in the form of rewards or penalties.

- **Classification vs. Regression:**
  Classification: Algorithms predict categorical or discrete class labels.
  Regression: Algorithms predict continuous numerical values.

- **Decision Tree-based vs. Neural Network-based:**
  Decision Tree-based: Algorithms build a tree-like model, making sequential decisions based on feature values.
  Neural Network-based: Algorithms consist of interconnected artificial neurons organized in layers, capable of learning complex patterns and relationships.

- **Linear vs. Non-linear:**
  Linear: Algorithms assume a linear relationship between inputs and outputs.
  Non-linear: Algorithms capture complex, non-linear relationships between inputs and outputs.

- **Batch Learning vs. Online Learning:**
  Batch Learning: Algorithms learn from a fixed dataset and require retraining on updated data to incorporate new information.
  Online Learning: Algorithms learn incrementally from streaming data, updating the model in real-time as new instances arrive.

It is interesting to note that these classifications are not mutually exclusive, and many machine learning algorithms can fall into multiple categories depending on their characteristics and usage. One might say that random forest falls into the supervised learning, decision tree based and classification algorithm categories.

2.0.2 Supervised and unsupervised learning

Supervised learning

Supervised learning is a machine learning approach where an algorithm learns from labeled data to make predictions or classifications. In supervised learning, a dataset is provided that consists of input variables (also called features or independent variables) and their corresponding output variables (also known as labels or dependent variables). The goal is to train a model that can generalize from the provided examples and accurately predict or classify unseen data.

The process of supervised learning involves two main steps: training and inference. During the training phase, the model is exposed to the labeled dataset, and it learns the underlying patterns and relationships between the input and output variables. This is typically achieved by minimizing a predefined loss function that measures the discrepancy between the model's predictions and the true labels.

The training data is divided into two subsets: a training set used to train the model and a validation set used to assess the model's performance during training and fine-tune its parameters. The model iteratively adjusts its internal parameters based on the feedback received from the loss function until it reaches an optimal state or convergence.

Once the model is trained, it can be used for inference on new, unseen data. The model takes the input variables as input and produces predictions or classifications as output. The performance of the model can be evaluated using various metrics, such as accuracy, precision, recall, or mean squared error, depending on the nature of the problem.

Supervised learning encompasses a wide range of algorithms, including linear regression, logistic regression, support vector machines (SVMs), decision trees, random forests, and neural networks. Each algorithm has its own assumptions, strengths, and limitations, making them suitable for different types of problems and datasets. In this thesis the zero age main sequence mass of the stars, the initial semimajor access of the binary star system, the initial eccentricity of the orbit, alpha and metallicity are used as the input parameters. Two output labels are also deduced which indicate if the binary star ends in a bound binary blackhole (BBH) and another label for merger BBH. In a more general way supervised learning used cases in astrophysics are:

- Galaxy Classification: Astrophysicists often study large datasets of galaxy images. Supervised learning algorithms can be trained on labeled images, where the morphological types of galaxies are known, to develop models that automatically classify

galaxies into different types (e.g., spiral, elliptical, irregular) based on their visual features.

- Exoplanet Detection: Exoplanets are planets orbiting stars outside our solar system. In exoplanet research, supervised learning can be used to analyze light curves obtained from telescopes and identify the characteristic patterns that indicate the presence of exoplanets. Labeled light curves with confirmed exoplanet detections can be used to train models for automated detection and classification of exoplanets.

- Supernova Classification: Supernovae are powerful stellar explosions and different types of supernovae exhibit distinct light curves and spectra. Supervised learning algorithms can be trained on labeled supernova data to classify and categorize supernovae based on their observed properties, helping to understand the underlying physical processes and improve their detection and classification efficiency.

Unsupervised learning

Unsupervised machine learning is a branch of machine learning where algorithms learn from unlabeled data without any explicit guidance or predefined labels. Unlike supervised learning, there is no ground truth or correct output to compare against during training. Instead, the algorithms aim to discover patterns, structures, or relationships within the data on their own.

In unsupervised learning, the algorithms explore the inherent characteristics of the data and group similar instances or identify underlying patterns without any prior knowledge. The main objectives of unsupervised learning include clustering, dimensionality reduction, and anomaly detection.

Clustering algorithms group similar data points together based on their intrinsic similarities or distances. The goal is to partition the data into meaningful clusters, enabling insights into the natural grouping of the data points. Examples of clustering algorithms include K-means, hierarchical clustering, and DBSCAN. A few used cases for unsupervised Learning in astrophysics, such as in Galaxy clustering - to analyze galaxy surveys and identify groups or clusters of galaxies with similar properties, Anomaly Detection in Astronomical Data - to identify rare or unusual events in astronomical data and Dimensionality Reduction in Spectroscopic Data - for preserving the relevant information after reducing the dimentionality of the data.

2.0.3 Random forest

Random forest is an ensemble learning algorithm that combines multiple decision trees to make predictions. It is a type of online learning method, where new data can be incrementally added to the model without retraining the entire forest. This makes random forest suitable for scenarios where data is continuously streaming or when updates need to be made to the model as new observations become available.
Ensemble learning is a powerful machine learning technique that combines multiple individual models (learners) to make predictions or decisions. Instead of relying on a single model, ensemble learning leverages the collective knowledge and diversity of multiple models to enhance prediction accuracy and generalization.Ensemble learning techniques come in various forms, including bagging, boosting, and stacking. Random forest, is an example of an ensemble learning method that utilizes bagging. Other ensemble methods, such as AdaBoost, Gradient Boosting, or XGBoost, operate based on the concept of boosting.
Decision trees are hierarchical models that represent a sequence of binary decisions and

outcomes. They are widely used in machine learning for both classification and regression tasks. The fundamental principles of decision trees include the recursive splitting of data based on features and thresholds.

The working of a decision tree can be broken down into the following

- Tree Structure: A decision tree consists of nodes and edges. The topmost node is called the root node, and it represents the entire dataset. Each internal node corresponds to a decision based on a specific feature and threshold.The edges emanating from an internal node represent the possible outcomes of the decision, leading to subsequent nodes or leaves. Leaves (also known as terminal nodes) represent the final prediction or outcome.
- Recursive Splitting: The process of building a decision tree involves recursively splitting the data based on features and thresholds to create decision rules. At each internal node, a feature and threshold are selected to partition the data into subsets. The goal is to find the feature and threshold that maximize the separation or purity of the subsets based on the target variable (in the case of classification) or the predicted value (in the case of regression).
- Splitting Criteria: The selection of the best feature and threshold is typically based on a splitting criterion, such as Gini impurity or entropy (for classification) or mean squared error (for regression). The splitting criterion measures the impurity or homogeneity of the subsets created by the split. The feature and threshold that result in the greatest reduction in impurity or error are chosen as the optimal split.
- Building the Tree: The process of building the decision tree continues recursively until a stopping criterion is met. This can be based on a maximum depth limit, a minimum number of samples required to split, or other predefined conditions. Each internal node represents a decision based on a feature and threshold, leading to subsequent nodes or leaves. The tree grows by iteratively adding nodes and splits until the stopping criterion is reached.
- Prediction: Once the decision tree is constructed, predictions can be made by traversing the tree from the root node down to a leaf node. At each internal node, the feature value of the input data is compared to the threshold. Based on the outcome, the traversal continues down the corresponding edge. When a leaf node is reached, the prediction is made based on the majority class (in classification) or the mean value (in regression) of the training examples in that leaf node.

Bagging, short for bootstrap aggregating, is a technique used in machine learning to improve the performance and robustness of models, particularly in the context of ensemble learning. Bagging is a fundamental component of random forest, where multiple decision trees are trained on different subsets of the data. The concept of feature randomness is an important aspect of random forest, contributing to its effectiveness in capturing diverse patterns and reducing the correlation between individual decision trees. In random forest, a random subset of features is considered at each node during the construction of each decision tree. The number of features in the subset is typically determined based on a user-defined parameter or a default value. The process of selecting the best feature and threshold for splitting is repeated recursively until a stopping criterion is met. Once all the decision trees in the random forest are constructed, their predictions are combined through majority voting (for classification) or averaging (for regression). The training process of

a random forest involves creating individual decision trees through bootstrapping and feature randomization. It has a few steps described above like bootstrapping, building Individual decision trees, feature randomization, aggregating predictions. This combination of techniques allows random forest to overcome over fitting, capture a wide range of patterns, and improve the predictive performance and generalization capabilities of the ensemble model.

The hyper parameters often considered for random forest include:

- Number of Trees (n_estimators): The number of trees in the random forest, denoted by n_estimators, is a crucial hyperparameter.Increasing the number of trees typically improves the performance of the random forest, but it also increases computational complexity.

- Maximum Depth of Trees (max_depth):The max_depth hyperparameter controls the maximum depth of each decision tree in the random forest. A deeper tree can capture more complex relationships in the data, but it may also lead to overfitting if not properly regulated.

- Minimum Samples Required for Splitting (min_samples_split): The min_samples_split hyperparameter determines the minimum number of samples required to split an internal node during tree construction. It ensures that a node is split only if it has a sufficient number of samples, preventing the creation of nodes with too few instances

The output class is predicted by the random forest by classification a new input, majority voting and confidence of the final decision. In particular in the classification of a new input, it is passed through each decision tree in the random forest. Each decision tree evaluates the input based on its learned rules and assigns it a class label. In majority voting, after all decision trees have made their predictions, the class labels and associated votes are counted. The confidence or probability is often estimated based on the proportion of decision trees in the random forest that predicted a particular class. The higher the proportion of trees predicting a specific class, the greater the confidence or probability associated with that class.

Random forest used for thesis

The thesis work provides a detailed account of the data, training, and validation process for two types of black hole mergers: bound BBH (Binary Black Hole) and merger BBH.

First, it introduces the concept of confusion matrices, which are used to evaluate the performance of the models. A confusion matrix provides a summary of the predicted and actual classifications of the data, indicating true positives, true negatives, false positives, and false negatives.

The importance of feature selection is emphasized in the thesis work, as it helps simplify the model and improve computational efficiency. Feature importance can be presented in various formats such as ranking the features based on their importance scores or visualizing them using graphical representations like bar plots or heatmaps. This allows for a clear identification of the most influential features in the dataset.

The analysis of feature importance offers several benefits in the context of the thesis work. It aids in understanding the relationships between features and the target variable, providing valuable insights for domain experts and guiding further investigations. It also helps in feature selection by excluding less important features, simplifying the model and improving efficiency. Furthermore, feature importance can detect potential biases or

spurious relationships in the data by assessing the impact of each feature on the model's predictions, identifying variables that may have a disproportionate influence or contribute to overfitting.

However, it is crucial to note that feature importance is specific to the model and dataset used in the thesis work. Different models may assign different levels of importance to features, and the results may vary depending on the evaluation metric employed. Therefore, it is essential to interpret and utilize feature importance within the context of the specific model and problem domain.

The thesis work then proceeds to discuss the challenges encountered during the training process with the SEVN data. One major challenge identified is the highly unbalanced nature of the data, where there is an insufficient number of data points representing one type of output. This imbalance can lead to overfitting and hinder the model's ability to recognize and classify certain types of events accurately.

To mitigate this challenge, the thesis work explores several techniques. One approach involves training the machine learning (ML) model with more features, such as incorporating the sum of Zams (Zero Age Main Sequence) mass of the stars, the pericenter and apocenter quantities derived from the semimajor axis and eccentricity, and the square of the semimajor axis.

Another technique involves training the merger BBH model using the data filtered specifically for bound BBH events. Since merger BBH events are a subset of bound BBH events, this approach aims to assess the performance of the ML model when trained with already filtered data.

Additionally, the thesis work examines the use of an already trained bounded BBH model to classify the data. This involves training an ML model for bound BBH data using the entire dataset, predicting the bound BBH values again using this model, and replacing the original values with the new predictions. The resulting dataset, consisting of predicted bound BBH and merger BBH data, is then used to retrain another ML model to detect merger BBH events.

The performance of each technique is evaluated using metrics such as confusion matrices and feature importance. The results show that none of the methods significantly improve over the initial method, indicating the complexity of the problem and the need for further investigation in the thesis work.

Finally, the thesis work explores the application of random grid search to optimize the performance of the ML algorithm, particularly the hyperparameters of a random forest. Random grid search involves identifying the current hyperparameters in use, specifying the range of values for each hyperparameter, and testing multiple combinations of hyperparameter values to find the best configuration. This technique can help improve the overall performance of the model.

In summary, the thesis work provides a detailed overview of the data, training, and validation

Chapter 3

SEVN

3.1 SEVN

3.1.1 Binary population synthesis code

Binary population synthesis codes represent the most efficient method for simulating the trajectory of binary star evolutionary (BSE), encompassing their journey from the zero-age main sequence (ZAMS) to their ultimate destiny. A notable example is the renowned BSE code ([6]Hurley et al., 2000, 2002), serving as the foundational progenitor for numerous binary population synthesis codes. This code can simulate the evolution of a staggering number of $O(10^6)$ binary stars within a matter of hours using a single CPU core. By comparison, contemporary stellar evolution codes necessitate approximately $O(10^2 - 10^3)$ CPU hours to model the evolutionary path of an individual binary star. The swift computational speed of binary population synthesis codes is not only crucial for exploring the parameter space of massive binary star evolution but also for facilitating seamless integration with dynamical codes, thereby enabling the study of the dynamic formation of BCOs (Binary Compact Objects) within densely populated stellar clusters. A multitude of binary population synthesis codes have been developed, contributing significantly to the investigation of Binary Compact Object (BCO) formation. These codes encompass are namely BINARY_C, BPASS, COSMIC, MOBSE, etc and SEVN [9]
Although each of these codes operates independently, a majority of them rely on a shared model of stellar evolution, which utilizes the accurate and computationally efficient fitting formulas developed by Hurley [6] et al. (2000). These fitting formulas enable the representation of key stellar evolution properties, such as photospheric radius, core mass, core radius, and luminosity, as functions of stellar age, mass (M), and metallicity (Z), representing the mass fraction of elements heavier than helium.
While binary population synthesis codes employing these fitting formulas may differ in their approaches to modeling stellar winds, compact-remnant formation, and binary evolution, they all rely on the same underlying stellar evolution model. Consequently, the outcomes obtained from these codes may vary, reflecting their divergent treatments of specific aspects, but ultimately relying on the common framework of stellar evolution.

3.1.2 SEVN

SEVN (Stellar EVolution for N-body) represents an efficient binary population synthesis code that facilitates swift calculations of stellar evolution. It achieves this by employing interpolation techniques on pre-computed collections of stellar tracks. Binary evolution is incorporated through the utilization of analytic and semi-analytic prescriptions. This

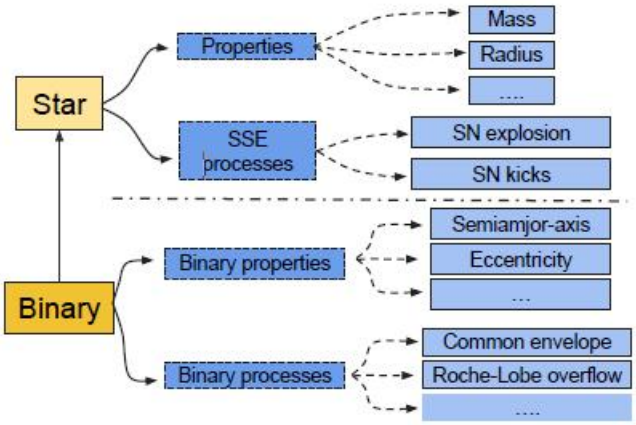


Figure 3.1: In SEVN, single stars (characterised by their properties like mass, radius,..) and binary stars(characterised by the properties like semi-major axis, eccentricity, ZAMS, ...). All these systems along with their properties are represented as C++ classes in the code

approach offers significant advantages as it enhances the code's generality and flexibility. The implementation of SEVN enables seamless modification or updating of the adopted stellar evolution models by simply loading a new set of look-up tables. Moreover, SEVN provides the capability to dynamically select the desired stellar tables during run time, without necessitating any internal code modifications or recompilation procedures.

SEVN is a C++ code developed following the object-oriented programming paradigm, designed to operate independently without relying on external dependencies. It leverages the advantages of CPU parallelization by utilizing OpenMP. Figure 3.1 provides a visual representation of the fundamental components of sevn and their inter-relationships.

Binary evolution

SEVN includes the binary evolution processes such as: wind mass transfer, Roche-lobe overflow (RLO), common envelope (CE), stellar tides, circularisation at the RLO onset, collision at periastron, orbit decay by GW emission, and stellar mergers. Binary-evolution prescriptions in SEVN rely on the BSE formalism (Hurley [6] et al. 2002); therefore, for binary evolution SEVN makes use of the BSE stellar types as mentioned in Table 3.1.

Table 3.1: SEVN stellar evolutionary phases (Column 0), identifiers (Column 1) and remnant types (Column 2). Column 3 shows the correspondence to Hurley stellar types:

SEVN Phase	Phase ID	SEVN Remanent subphase	Remanent ID	BSE Stellar-type equivalent
Pre-main sequence (PMS)	0	-	0	not available
Main sequence	1	-	0	1 if $f^{\dagger}_{conv}$ ¡ 0.8, else 0
Terminal-age main sequence (TAMS)	2	-	0	2 if $f^{\dagger}_{conv}$ ¡ 0.33, else 3
Shell H burning(SHB)	2	-	0	
Core He burning (CHeB)	2	-	0	7 if $WR^{\ddagger}$ else 4
Terminal-age core He burning(TCHeB)	5	-	0	7 if $WR^{\ddagger}$ else 4 if $f^{\dagger}_{conv}$ ¡ 0.33, else 5
Shell He burning (SHeB)	6	-	0	8 if $WR^{\ddagger}$ else 4 if $f^{\dagger}_{conv}$ ¡ 0.33, else 5
Remnant	7	He white dwarf(HeWD)	1	10
		CO white dwarf(COWD)	2	11
		ONe white dwarf(ONeWD)	3	12
		neutron star formed via electron capture (ECNS)	4	13
		neutron star formed via core collapse (CCNS)	5	13
		blackhole (BH)	6	14
		no compact remnant(Empty)	-1	15

1. Wind mass transfer
   In SEVN, the stellar tracks stored in the tables are assumed to incorporate wind

mass loss, ensuring that this aspect is consistently considered in the single stellar evolution. Additionally, we account for the potential scenario where mass and angular momentum lost from a star (referred to as the donor) can be accreted by its stellar companion (known as the accretor).The stellar tracks stored in the tables are assumed to incorporate wind mass loss, ensuring that this aspect is consistently considered in the single stellar evolution. Additionally in the paper[8], they have accounted for the potential scenario where mass and angular momentum lost from a star (referred to as the donor) can be accreted by its stellar companion (known as the accretor).

$$\frac{\dot{a}}{a} = \frac{\dot{M}_d}{M_a + M_d} - (\frac{2 - e^2}{M_a}\frac{1 + e^2}{M_a + M_d})\frac{\dot{M}_a}{1 - e^2} \tag{3.1}$$

and

$$\frac{\dot{e}}{e} = -\dot{M}_a[(M_a + M_d)^{-1} + 0.5M_a^{-1}] \tag{3.2}$$

The mass loss through stellar winds leads to an expansion of the orbit. However, the mass accretion onto the companion star counteracts this effect by restoring some of the lost angular momentum to the system (Equation (3.1)). Furthermore, the accretion of wind mass reduces the eccentricity of the orbit, resulting in circularization (Equation (3.2)). It is important to note that these variations in eccentricity, even during the periods of intense mass loss, are insignificant compared to those induced by stellar tides.

2. Roche-lobe overflow
Within the SEVN code, a Roche lobe overflow (RLO) initiates when the radius of either star reaches or exceeds the value of R_L. The RLO process continues until this criterion is no longer met or until mass transfer results in a merger or a common envelope (CE) event. At each time-step, SEVN actively verifies the fulfillment of this condition.

$$R_L = \frac{0.49q^{2/3}}{0.6q^{2/3} + ln(1 + q^{1/3})} \tag{3.3}$$

3. Stability criterion
The process of Roche lobe overflow (RLO) brings about alterations in the mass ratio, masses, and semi-major axis of the binary system. Consequently, the size of the Roche lobe (RL) undergoes contraction or expansion (described by Equation (3.3)). If the RL contracts at a faster rate than the donor star's radius (or if the RL expands slower than the donor's radius) due to the star's adiabatic response to mass loss, the mass transfer becomes dynamically unstable. This instability occurs on a timescale comparable to the system's dynamical timescale and can lead to either a stellar merger or the formation of a common envelope (CE) configuration. From the paper [8] "population synthesis codes usually implement a simplified formalism in which the mass transfer stability is evaluated by comparing, $q = M_d/M_a$ (M_d = Mass of donor star;l M_a = Mass of the accretion star) with some critical value q_c. If the mass ratio is larger than q_c,the mass transfer is considered unstable on a dynamical time scale.The critical mass ratio is usually assumed to be large (¡ 2) for stars with radiative envelopes (e.g., MS stars, stars in the Hertzsprung-gap phase, and pure-He stars),while it is smaller for stars with deep convective envelopes (but see Ge et al. 2020b [4],a[5], for a significantly different result)."

4. Stable Mass transfer
   In the updated version of sevn, they have incorporated a slightly modified formalism to describe stable mass transfer, taking into account differences compared to the approaches of Hurley [6] et al. (2002) and Spera et al. (2019). The main distinctions are outlined below. The rate at which mass is lost depends on the degree to which the donor star exceeds the size of its Roche lobe (as described by Hurley [7] et al., 2002):

$$\dot{M}_d = -F(M_d)\left(\ln\frac{R_d}{R_L}\right)^3 M_\odot\,\mathrm{yr}^{-1} \tag{3.4}$$

   and the normalization factor is

$$F(M_d) = 3\times10^{-6}\left(\min[M_d, M_{\mathrm{max,SMT}}]\right)^2 \times \begin{cases} \max\left[\frac{M_{\mathrm{env,d}}}{M_d}, 0.01\right], & \text{for HG phase donors} \\ 10^3 M_d\left(\max[R_d, 10^{-4}]\right)^{-1}, & \text{for WD donors} \\ 1, & \text{all other cases} \end{cases} \tag{3.5}$$

   where all the quantities are in solar units.

5. Orbital variations In the case of a non-conservative mass transfer ($MT \neq 1$), there is a loss of angular momentum from the system. We parameterize this angular momentum loss as follows:

$$\Delta J = -|\Delta M_{loss}|\gamma_{RLO}a^2\sqrt{1-e^2}\frac{2\pi}{P} \tag{3.6}$$

where P is the orbital period and ΔM is the actual mass lost from the system in a given evolution step, i.e. the difference between the mass lost by the donor and that accreted on the companion[8]. Therefore, the spin angular momentum lost by the donor is added to the orbital angular momentum:

$$\Delta J_{orb,d} = -\Delta J_{spin,d} = -\Delta M_d R_L^2 \Omega_{spin,d} \tag{3.7}$$

where ΔM_d is the mass lost by the donor in an evolutionary step and $\Omega_{spin,d}$ is the donor angular velocity.In contrast, the accretion of mass onto the companion star results in the removal of orbital angular momentum and an increase in the spin of the accretor.

$$\Delta J_{orb,a} = -\Delta J_{spin,a} = -\Delta M_a\sqrt{GM_aR_acc} \tag{3.8}$$

Finally, the change in the semi-major axis resulting from the Roche lobe overflow (RLO) is estimated as follows:

$$\Delta a = \frac{(J_{orb} + \Delta J_{orb,lost} + \Delta J_{orb,a})^2(M_a + J_{orb} + M_d)}{G(1-e^2)M_d^2M_a^2} \tag{3.9}$$

where the masses are considered after the mass exchange in the current time-step. Similarly, the variations in stellar spins are updated based on Equations (3.7) and (3.8). The are other process considered is Unstable mass transfer, for the present model of SEVN.

Common envelope

The common envelope (CE) phase is a distinctive stage in the evolution of a binary system, characterized by the binary being enveloped within the expanded envelope of one or both of its components. During this phase, the loss of synchronization between the binary orbit and the envelope leads to drag forces that cause the orbit to shrink, while the envelope gains energy and expands. This formulation relies on comparing the energy required to disperse the stellar envelope(s) with the orbital energy before and after the common envelope (CE) event. The calculation of these two energy terms is contingent upon two parameters: $\lambda_{CE} and \alpha_{CE}$ The initial parameter, λ_{CE}, serves as a structural parameter that determines the binding energy of the stellar envelope (as described by Hurley[7] et al., 2002). Consequently, the binding energy of the common envelope can be expressed as:

$$E_{bind,i} = -G\left(\frac{M_1 M_{env1}}{\lambda_{CE1} R_1} + \frac{M_2 M_{env2}}{\lambda_{CE2} R_2}\right) \tag{3.10}$$

The parameter α_{CE} denotes the fraction of orbital energy that is converted into kinetic energy of the envelope throughout the common envelope evolution. Thus, the energy required to unbind the envelope can be expressed as:

$$\Delta E_{orb} = \alpha_{CE} \frac{G M_{c,1} M_{c,2}}{2} (a_f^{-1} - a_i^{-1}) \tag{3.11}$$

where $M_{c,1}$ and $M_{c,1}$ are the core masses of two stars and $a_f and a_i$ are the final and initial semi-major axis after and before the CE phase. For stars that lack a well-defined envelope (such as main-sequence stars, pure-He stars without a CO core, naked-CO stars, and compact remnants), we assume $E_{bind} = 0$ and $M_c = M$. To determine the post-common envelope separation, we enforce $E_{bind,i} = \Delta E_{orb}$. If neither star fills its Roche lobe in the post-common envelope configuration, we assume that the common envelope is ejected. Otherwise, the two stars merge.

The labels "star 1" and "star 2" are used solely to differentiate between the two merging objects and do not necessarily refer to the primary and secondary stars. It is important to note that a type Ia supernova (SNIa) does not result in the formation of a compact remnant.

Gravitational waves

SEVN describes the impact of GW emission on the orbital elements by including the same formalism as BSE [8]:

$$\dot{a} = -\frac{64 G^3 M_1 M_2 (M_1 + M_2)}{5 c^5 a^4 (1 - e^2)^{7/2}} \left(1 + \frac{73}{24} e^2 + \frac{37}{96} e^4\right) \tag{3.12}$$

$$\dot{e} = -\frac{304 G^3 M_1 M_2 (M_1 + M_2)}{15 c^5 a^4 (1 - e^2)^{5/2}} \left(1 + \frac{121}{304} e^2\right) \tag{3.13}$$

The aforementioned equations, originally presented by Peters [11] (1964a), incorporate the effects of gravitational waves (GWs) on orbital decay and circularization. Unlike in the BSE code, where Equations 39 and 40 are active only for semi-major axes less than 10 AU, in sevn these equations are activated whenever the GW merger timescale, t_{merge}, is shorter than the Hubble time. The GW merger timescale is evaluated using a highly accurate approximation of the solution to the systems of Equations 39 and 40, ensuring errors of less than 0.4%.

	ID	name	Mass_0	RemnantType_0	Mass_1	RemnantType_1	Semimajor_final	Eccentricity_final	totalDelay	boundedBBH	mergerBBH	alpha	metallicity
1	70000	399815311357436	14.158780	6	0.839984	1	356.5665	0.000000	1.358539e+10	0	0	1.0	0.002
3	70001	958183953698478	35.298340	6	30.194140	6	3322.4210	0.080821	2.557707e+11	1	0	1.0	0.002
5	70002	760104814589280	1.009558	2	NaN	-1	NaN	NaN	NaN	0	0	1.0	0.002
7	70003	781505964375302	1.235519	5	0.901158	2	NaN	NaN	NaN	0	0	1.0	0.002
9	70004	276199685952483	0.898976	2	NaN	-1	NaN	NaN	NaN	0	0	1.0	0.002
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2001991	1000095	326946239407856	1.235519	5	NaN	-1	NaN	NaN	NaN	0	0	0.1	0.020
2001993	1000096	831682635041421	1.331054	5	NaN	-1	NaN	NaN	NaN	0	0	0.1	0.020
2001995	1000097	255883826177977	1.235519	5	1.064472	2	NaN	NaN	NaN	0	0	0.1	0.020
2001997	1000098	764405085484896	1.008198	2	0.804301	2	225508.2000	0.046510	2.618048e+23	0	0	0.1	0.020
2001999	1000099	251848342117327	0.903298	2	NaN	-1	NaN	NaN	NaN	0	0	0.1	0.020

9009000 rows × 13 columns

Figure 3.2: Shows the data frame containing the final and initial values of the binary system

3.2 Data, training and validation

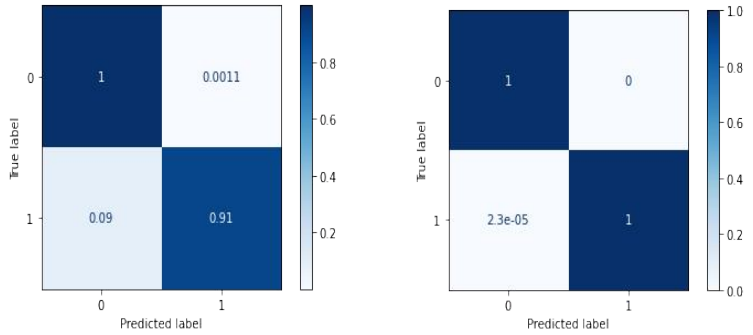
The data from several SEVN simulation runs are put together to be used to train the random forest algorithm. The different runs of SEVN have different values of α (fraction of binding energy used to expel the common envelope) value from (1.0,10.0,100.0) and different metallicities like (0.002,.0002,0.00002). Hence in total it gives us number of different combinations of input parameters for the runs.
These different runs of SEVN each give out a set of comma separated value files (.csv) which are individually processed, around 20 files. All of these files in total will be around 180 files. These files are together put into a single data frame, so as to make it easier to manipulate. In the parameter list we take only the zero age main sequence of the two stars (Zams_0,Zams_1), initiail semi major axis of the star system (Semimajor_initial), initial eccentricity of the orbit(Eccentricity_initial),metallicity*(Z) and α_{CE}, other parameters are dropped from the input data.
Next we create two more columns to hold to filter the data and determine if a binary star system eneded up to be a bound BBH or a merger BBH. If the system has a both the stars ending with a phase value of 6, that is the stars ended up as remnants at the end of the simulation and remnant type of 7, which is the one for blackhole, only such data rows are considered. In rows which satisfy these conditions, we check on if the GWTime is valid or not NA, then we identify it as bound BBH system and make the column 1. We further take the ones which satisfy these conditions and check if the sum of simulation time ('BWorldTime') and GWtime is less than 14 Billion years, a sample of this prepared data frame is given in figure 3.2, then these data rows will be classified as merger BBH as.

3.2.1 Training the model

Hence the data to be used for training the model (Zams_0,Zams_1), initiail semi major axis of the star system (Semimajor_initial), initial eccentricity of the orbit(Eccentricity_initial),metallicity and α,as the input parameters for training the model(in our case random forest). The 'boundedBBH' and 'mergerBBH' as the output parameters for the model. The package we use sklearn.ensemble to import RandomForestClassifier and train the random forest model. We use the package train_test_split from sklearn.model_selection, for splitting the data from previous section to training data and testing data. Lets call this the method 1, in which we take 6 featuers as input and train for bound and merger BBH Training the model takes around 10 minutes with 6 million data points, with an average of 120,000 bound merger BBH systems and 2600 merger BBH systems. The confusion matrix on training the data with bound BBH is as follows, it shows a little bit of overfitting having occured! Same is the case with the merger BBH data as well. This is probably because of the distribution of data points, the data we have used to train the model is highy unbalanced, the overfitting

```
[ ] X = wholeDataBBH[['Zams_0','Zams_1','Semimajor_initial','Eccentricity_initial','alpha','metallicity']]
    y_bounded = wholeDataBBH[['boundedBBH']]
    y_merger = wholeDataBBH[['mergerBBH']]

[ ] X_train, X_test, y_train, y_test = train_test_split(X, y_bounded, test_size=0.3, random_state=11) # 70% training and 30% test
    # random_state=11 for keeping the same combo of test and train sets for all models
```

Figure 3.3: Code giving the peak into picking data for input and output labels for training the model, as well as splitting the data into training and testing



(a) Confusion matrix on test data

(b) Confusion matrix on training data

Figure 3.4: Gives the confusion matrix for bound BBH, for training and test data

can be avoided by a number of techniques such as synthetically increasing the number of merger and bound BBH positive cases.

Confusion matrix

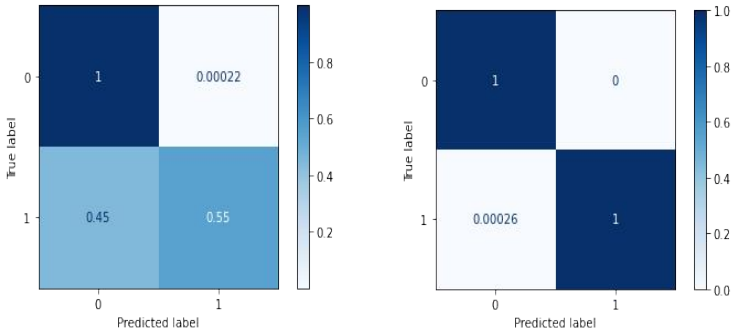
The confusion matrix is a fundamental tool in evaluating the performance of a classification model. It provides a tabular representation of the predicted and actual class labels for a given dataset. The matrix is typically square and organized into four quadrants: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

The TP quadrant represents the cases where the model correctly predicted a positive class, and the actual class is indeed positive. TN represents the cases where the model correctly predicted a negative class, and the actual class is indeed negative. FP corresponds to the cases where the model incorrectly predicted a positive class when the actual class is negative. FN represents the cases where the model incorrectly predicted a negative class when the actual class is positive.

The confusion matrix provides valuable information to assess the performance of a classification model. From the matrix, several performance metrics can be derived, such as accuracy, precision, recall, and F1 score. These metrics help in understanding the model's ability to correctly classify instances and identify any potential biases or weaknesses.

Analyzing the confusion matrix allows for a deeper understanding of the model's strengths and limitations. It can reveal patterns of mis-classifications, highlighting specific classes that are prone to errors or instances where the model excels. By examining the distribution of predictions across the matrix, insights can be gained into the model's overall effectiveness and potential areas for improvement.

The confusion matrix enables the evaluation of different classification algorithms or parameter settings. It facilitates the comparison of models based on their performance in differentiating between classes and provides a basis for selecting the most suitable approach for a specific task.



(a) Test data (b) Training data

Figure 3.5: Gives the confusion matrix for merger BBH

The confusion matrix for merger BBH data also shows some degree of over fitting having occurred.

Feature importance

Feature importance can be presented in different formats, including ranking the features based on their importance scores or visualizing them using bar plots, heatmaps, or other graphical representations. This allows for a clear identification of the most influential features in the dataset.

The analysis of feature importance provides several benefits. It helps in feature selection, where less important features can be excluded to simplify the model and improve computational efficiency. It also aids in understanding the underlying relationships between features and the target variable, providing valuable insights for domain experts and guiding further investigations.

Furthermore, feature importance can assist in identifying potential biases or spurious relationships in the data. By assessing the impact of each feature on the model's predictions, it is possible to detect variables that may have a disproportionate influence or contribute to overfitting.

However, it is essential to note that feature importance is specific to the model and dataset used. Different models may assign different levels of importance to features, and the results may vary depending on the evaluation metric employed. Therefore, it is crucial to interpret and utilize feature importance within the context of the specific model and problem domain.

The feature importanace for merger BBH and bound BBh are as in the figure 3.6 and 3.7. In both clearly the Zams mass of both stars looks like the most important feature, next is semi major axis and then eccentricity.

3.2.2 Challenges encountered while training with SEVN data

When training with the SEVN data for both bound BBH and merger BBH, the fact that the data was highly unbalanced was the one which resulted in over fitting and also as there are not enough number of data points representing one kind of output, the ML might not get trained sufficient enough to recognize this. As an example in our data set, out of the 6178148 data points used in bound BBH only 128149 of them resulted in actual bound BBH, that is 2.07% of data represneted the features which gave bound BBH. Similarly in case of merger BBH, out of 2645211 only 49716 number of points resulted in merger BBh, which is 1.88% . In both cases it is clear that the cases for detecting the BBH are

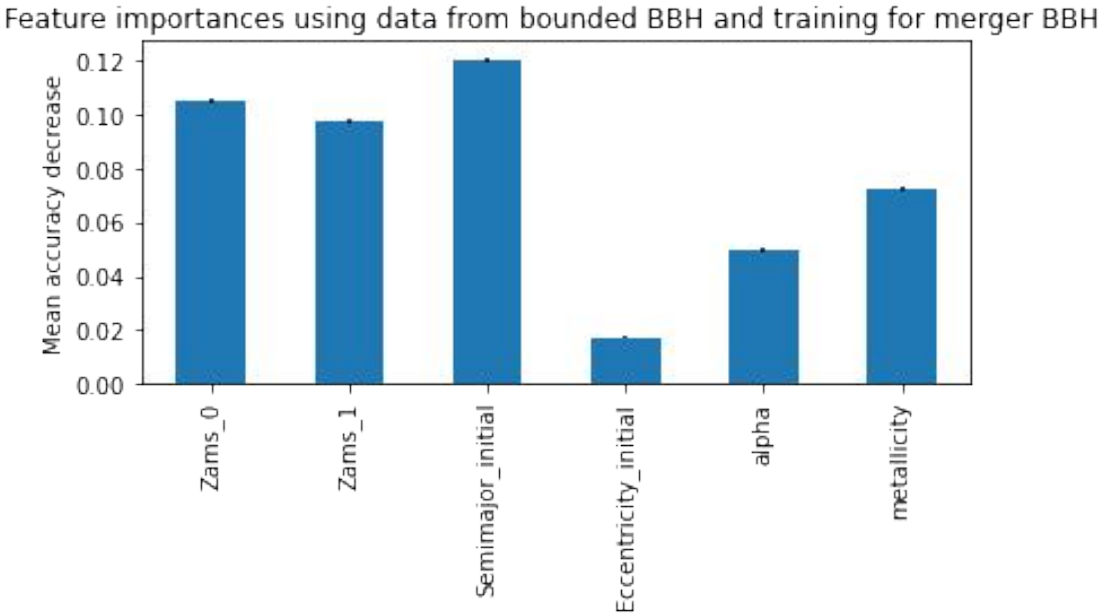


Figure 3.6: Here is the feature importance for the merger BBH, giving us a vague idea that Zams mass is very important, next is semi major axis and eccentricity

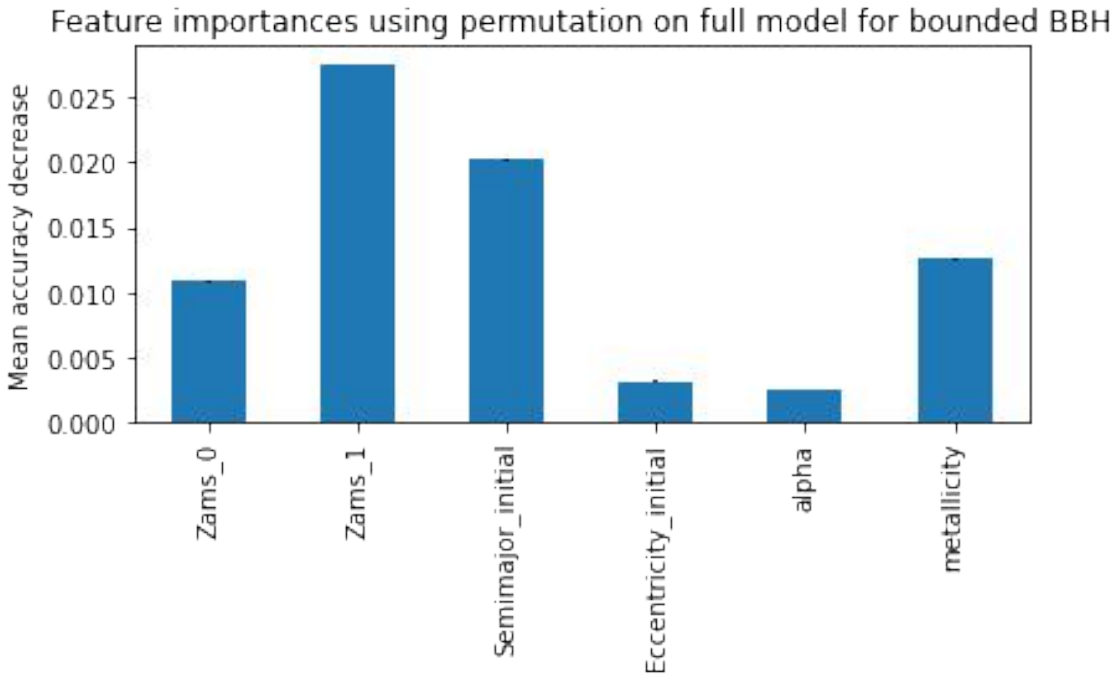


Figure 3.7: Here is the feature importance for the Bound BBH

```
X_withAllFeatures = wholeDataBBH[['Zams_0','Zams_1','Semimajor_initial','Eccentricity_initial','alpha','metallicity']]
X_withAllFeatures.loc[:,'SumOfZams'] = X['Zams_0'] + X['Zams_1']
X_withAllFeatures.loc[:,'Pericenter'] = X['Semimajor_initial']*(1- X['Eccentricity_initial'])
X_withAllFeatures.loc[:,'Apocenter'] = X['Semimajor_initial']*(1+ X['Eccentricity_initial'])
X_withAllFeatures.loc[:,'SemimajorSq'] = X['Semimajor_initial']*X['Semimajor_initial']

X_train_withAllFeature, X_test_withAllFeature, y_train_withAllFeature, y_test_withAllFeature = train_test_split(X_withAllFeatures
```

Figure 3.8: Train/ test splitting of the data for training with all the features plus extra ones described above

not enough and we need to find ways of mitigating this, we have used several techniques or tricks to achieve this. The following are some of them we have employeed and tested, using confusion matrix and feature importance as two parameters to check the performance, also noting the computation time:

Training with more features(method 2)

Here we increase the number of features we train the ML on, namely we add, referring to the code 3.8

- Sum of Zams: taking another feature which is the sum of Zams0 and Zams1 of the stars. Using the command
- Pericenter : considering the equation

$$R_p = a(1 - e)$$

  where R_p is the pericenter, a is semimajor axis and e = eccentricity. Taking this quantity R_p as a feature

- Apocenter: considering the equation

$$R_a = a(1 + e)$$

  where R_a is the apocenter, a is semimajor axis and e = eccentricity. Taking this quantity R_a as a feature

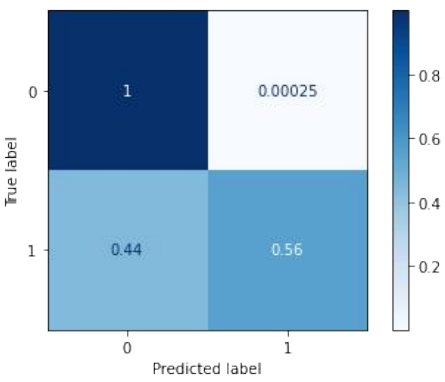
- Square of the semimajor axis: just squaring the semimajor axis and taking it as a feature

Results for method 2: The feauture importance and confusion matrix do not show any significant improvement over method 1, keeping the computation time as well the same 636 second for method 1 and 639 seconds for method 2.

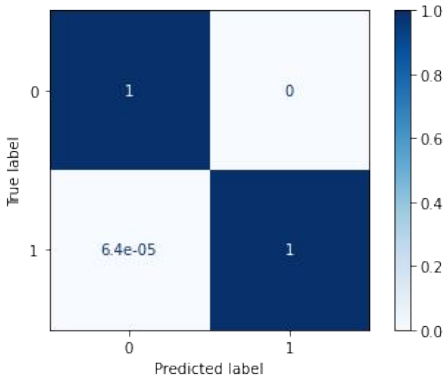
Training merger BBH based on bound BBH data (method 3)

Here we use the data from the whole data frame as in mthod one with an additional level of filtering for only the bound BBH and using this filter data to train the merger BBH. We know that the merger BBh events are a subset of bound BBH events, so we want to check the performance of the ML to detect merger BBH but when the training data contains already filtered data.3.10.

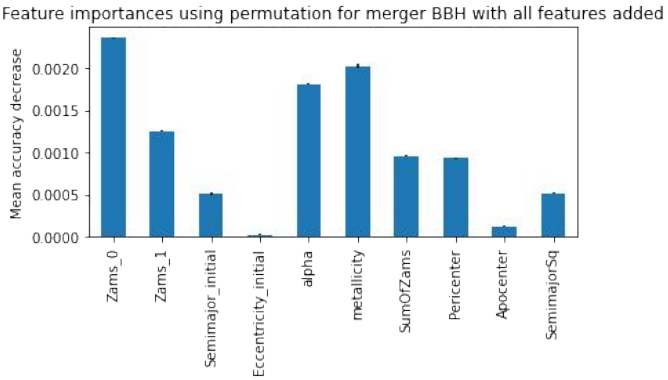
Results for method 3: The feauture importance and confusion matrix do not show any significant improvement over method 1, while the computation time improved a lot 636 second for method 1 and 26.542 seconds for method 3, as the number of data points for training has significantly reduced.



(a) Confusion matrix for test data



(b) Confusion matrix for training data set

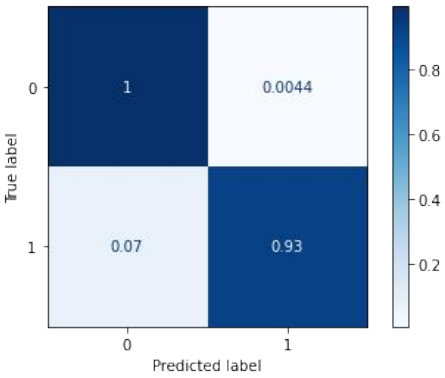


(c) Metrics for checking the performance of method 2 training

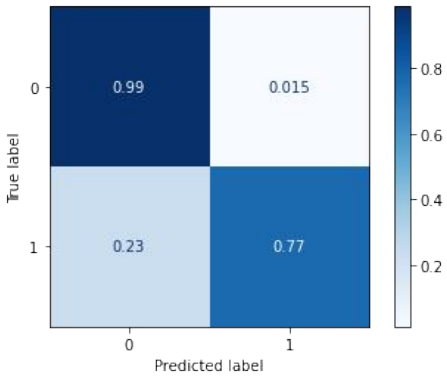
Figure 3.9: Metrics for checking the performance of method 2 training

```
boundedData = wholeDataBBH[ wholeDataBBH['boundedBBH'] == 1]

X_bounded2merger = boundedData[['Zams_0','Zams_1','Semimajor_initial','Eccentricity_initial','alpha','metallicity']]
y_bounded2merger = boundedData['mergerBBH']
X_train_boundAndMerger, X_test_boundAndMerger, y_train_boundAndMerger, y_test_boundAndMerger = train_test_split(X_bounded2merger
# 70% training and 30% test

clf_boundAndMerger=RandomForestClassifier(n_estimators=100)

#Train the model using the training sets y_pred=clf.predict(X_test)
clf_boundAndMerger.fit(X_train_boundAndMerger,y_train_boundAndMerger)
```

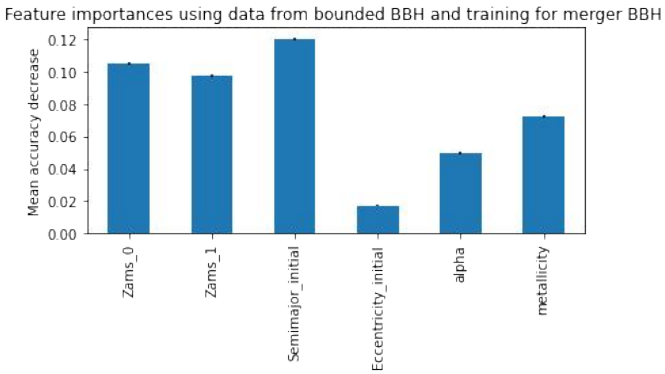
Figure 3.10: Train/ test splitting of the data for training by using filter for bound BBH ones described above



(a) Confusion matrix for bound BBH



(b) Confusion matrix for merger BBH



(c) Metrics for checking the performance of method 3 training

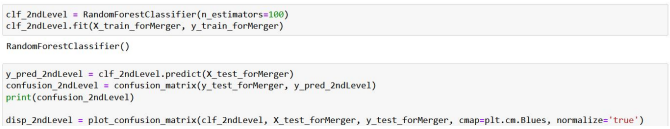
Figure 3.11: Metrics for checking the performance of method 3 training

Using already trained Bounded BBH model to classify data (method 4)

Here we use the data from the whole data frame to train a ML model for bound BBH data, then we use this model to predict the bound BBH values again and replace the original bound BBH values with the new prediction. Next we use this new data frame of merger BBH and predicted bound BBH, to retrain yet another ML to check for merger BBH3.12

(a) Training with all of the data for bound BBH

(b) Predicting bound BBH



(c) 2nd level training ML for merger BBH

Figure 3.12: Commands for training, predicting for bound BBH and retraining for merger BBH

Results for method 4: The feauture importance and confusion matrix do not show any significant improvement over method 1, keeping the computation time as well the same 636 second for method 1 and 639 seconds for method 2.

Random grid search

Another method to optimize the performance of a ML algorithm is to find the best hyper parameters for ML used, in this case the hyper parameters of a random forest. As a first step, the hyper parameters currently in use for the random forest need to be noted down, as in the figure below3.13. A well known technique to search for the best hyper parameters is known as random grid search Next the selection of values each hyper parameter takes can be noted, based on the previously got list of values, for example in case of max_depth which had a value of 'None; or was kept empty before can take values like $'max\_depth' : [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None]$. In this way we can give various ranges or set of values for each hyper parameter. A complete list of all values given to all the hyper parameters is given in the figure3.14. this selection of values will give us a total of 100 combinations to be tested for the grid search As the next step, the RandomizedSearchCV is command used from the *klearn.model_selectio* package. The RandomizedSearchCV will execute, the random forest ML with cross validation level of 3, so in total 300 iterations it will go through as in figure 3.15. This command might run for a significant amount of time, in my case it took around 12 hours to finish, finally it will give us a list of the best hyper parameters to use for the random forest, as in the figure below 3.16

```
# Look at parameters used by our current forest
from pprint import pprint
print('Parameters currently in use:\n')
pprint(clf_2ndLevel.get_params())
```

```
Parameters currently in use:

{'bootstrap': True,
 'ccp_alpha': 0.0,
 'class_weight': None,
 'criterion': 'gini',
 'max_depth': None,
 'max_features': 'auto',
 'max_leaf_nodes': None,
 'max_samples': None,
 'min_impurity_decrease': 0.0,
 'min_impurity_split': None,
 'min_samples_leaf': 1,
 'min_samples_split': 2,
 'min_weight_fraction_leaf': 0.0,
 'n_estimators': 100,
 'n_jobs': None,
 'oob_score': False,
 'random_state': None,
 'verbose': 0,
 'warm_start': False}
```

Figure 3.13: This is an exhaustive list of hyper parameters and their values currently in use in the random forest.

```
from sklearn.model_selection import RandomizedSearchCV
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}

pprint(random_grid)
```

```
{'bootstrap': [True, False],
 'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None],
 'max_features': ['auto', 'sqrt'],
 'min_samples_leaf': [1, 2, 4],
 'min_samples_split': [2, 5, 10],
 'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]}
```

Figure 3.14: This is selection of hyper parameters and their values which are in use in the random forest, for the random grid search.

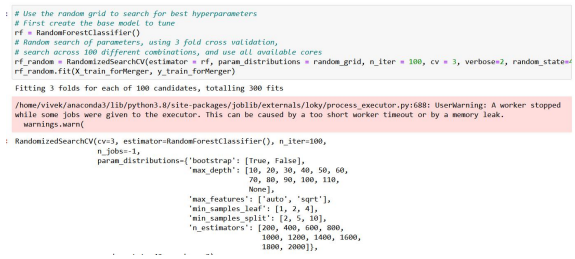


Figure 3.15: Actual code for the random search cv.

```
rf_random.best_params_
```

```
{'n_estimators': 1400,
 'min_samples_split': 2,
 'min_samples_leaf': 1,
 'max_features': 'auto',
 'max_depth': 40,
 'bootstrap': False}
```

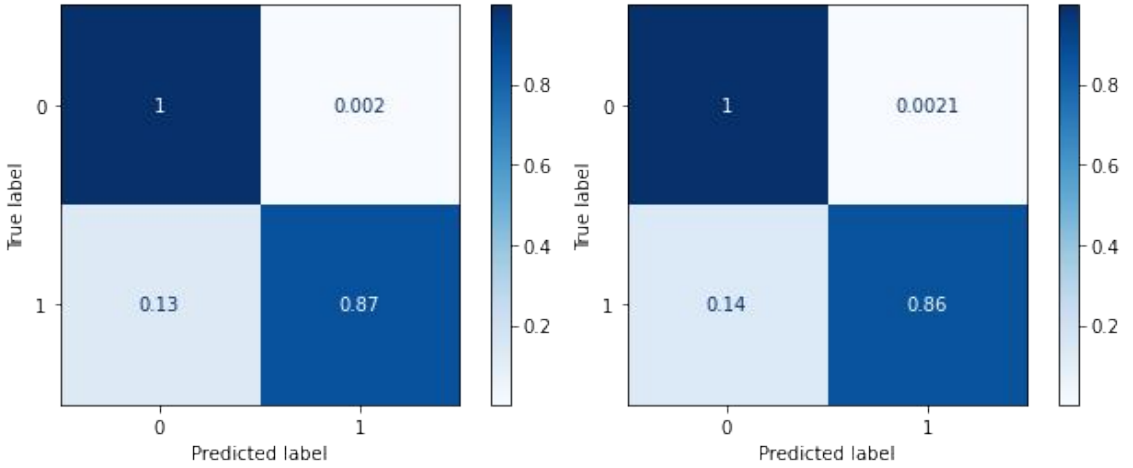
Figure 3.16: Final result for the random search cv giving the best hyper paramters we can use.

XGBoost : method 5

XGBoost (Extreme Gradient Boosting) is a powerful and widely-used machine learning algorithm known for its exceptional performance in various predictive modeling tasks. It belongs to the family of ensemble learning methods, which combine the predictions of multiple models to achieve better overall accuracy. XGBoost leverages the concept of gradient boosting and introduces several innovations to enhance its effectiveness and efficiency. By employing a combination of decision trees and a gradient-based optimization framework, XGBoost can effectively handle complex feature interactions, handle missing data, and handle a large number of features. Its ability to capture non-linear relationships, handle high-dimensional data, and provide feature importance rankings makes it a popular choice in competitions and real-world applications. With its impressive track record and widespread adoption, XGBoost continues to be a valuable tool for data scientists and machine learning practitioners seeking state-of-the-art predictive modeling capabilities.

Here we take the same parameters as we did for method 1 where we take zams 0, zams 1 of the two stars, initial eccentricity, initial semimajor axis, α and metallicity as the input parameters and then bound BBH or merger BBH column as output. Each time we train with bound BBH, we drop that column and then train, splitting the data into 70% for training and 30% for testing. Same is the case with merger BBh.

Results for method 5: The computation time for training with XGBoost for bound BBH is approximately 38 seconds and the same for merger BBH is 39 seconds, this shows a significant reduction in the computation time, by around 20 times compared to method 1 of random forest. But the confusion matrix for all cases shows that the performance for XGBoost is much better than random forest but with reduced computation time, in case of both bound BBh and merger BBH. So it is a better choice for using the SEVN data with.Both the figures 3.19 and 3.17 show similar results and confusion matrix infomation.



(a) Training with all of the data for merger BBH, with XGBoost, confusion matrix

(b) XGBoost testing merger BBH

Figure 3.17: Metrics for checking the performance of XGBoost training with merger BBH

XGboost with two level ML : method 6

We again repeat what we did in method 4, in that we train the model first for bound BBH but this time using XGboost and then we use this trained model to predict Bound BBH column again and then we train the data based on this new data frame for merger BBH. Here again there is drastic reduction in computational time while training for both bound and merger BBH.

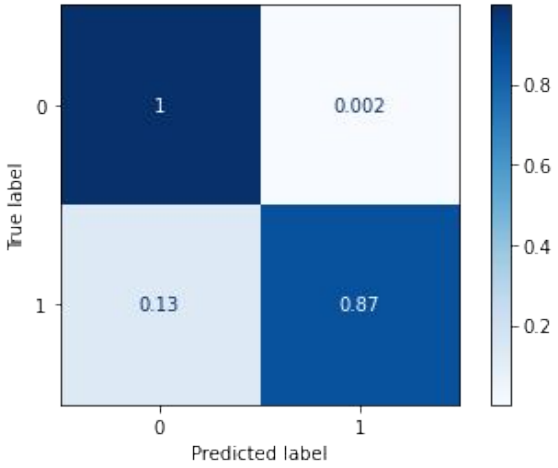
Using SMOTE : method 6

Oversampling the minority class by duplicating examples or synthesizing new ones is a common solution. Synthetic Minority Oversampling TEchnique (SMOTE) is a widely used method that generates synthetic examples by connecting random instances of the minority class and creating new samples along that line. This approach improves class balance and helps classifiers build larger decision regions for the minority class. However, it may result in ambiguous examples if there is a strong overlap between the majority and minority classes. Combining SMOTE with undersampling of the majority class has shown better performance compared to undersampling alone.

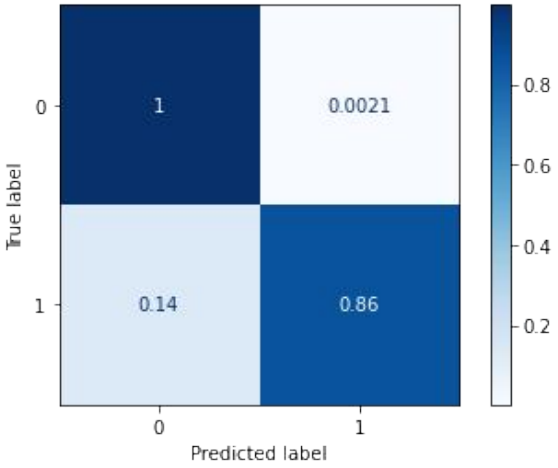
Here we can apply SMOTE technique so as to oversample bound BBH or merger BBH cases, so as to make these classes more and thus increase the probability of the ML picking up on key featuers of these two classes.

A sample of how the original data is with unbalanced data classes is given in the figure. These graphs show some sample data which is highly imbalanced, they are simulated data as given in the website. These are generated using random Rand function, lets assume that the x-axis is a dummy parameter X which is independent and y-axis is a dependent parameter Y, in the following figures.

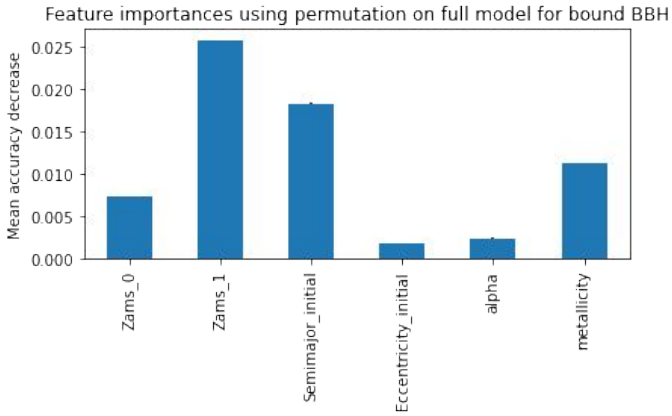
After applying SMOTE, in two different methods, 1st method the number of cases for the lesser represented class is made same as that for the majority class 3.21. For the 2nd method, the number of cases for majority class to minority class is 2:1, as shown in the



(a) Training with all of the data for bound BBH, with XGBoost, confusion matrix

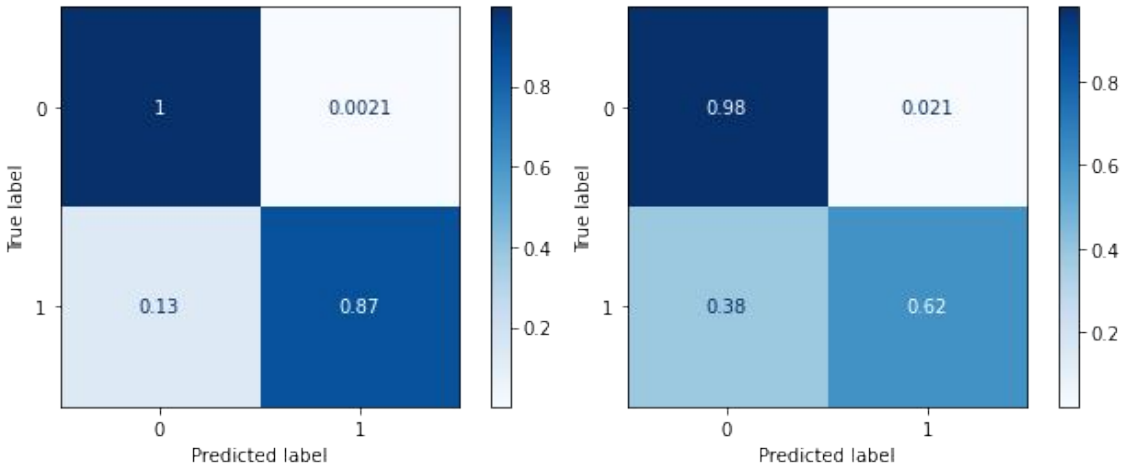


(b) XGBoost testing bound BBH



(c) XGboost bound BBH data feature importance

Figure 3.18: Metrics for checking the performance of XGBoost training with bound BBH figure below . This SMOTE technique can be applied to our data set and retrain the ML on the new synthetic data and this results might improve our ML. This work can be noted as the future improvements to this present body of work.



(a) Training with all of the data for bound BBH and after retraining model based on prediction made by XGBoost

(b) XGBoost testing merger BBH but second level, ie, using this to create column of predicted BBH, with the first level XGBoost algo was training

Figure 3.19: Metrics for checking the performance of XGBoost training first level of training with bound BBH and then second level of training with prediction of bound BBH values from first level

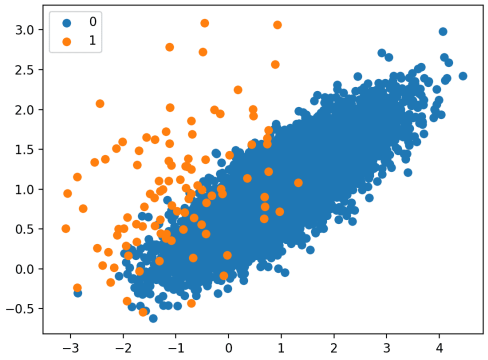


Figure 3.20: Here is the sample unbalanced data set, taken from the website `https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/`. With x-axis = X, y-axis = Y, dummy parameters

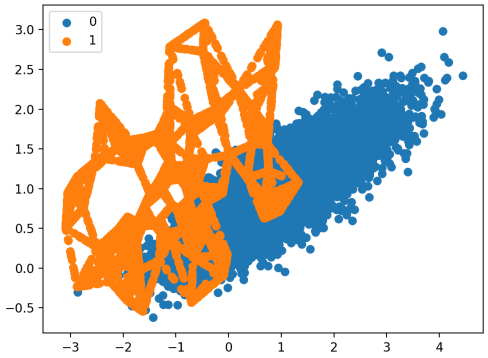


Figure 3.21: Number of cases for lower represented class is same as for majority classes.With x-axis = X, y-axis = Y, dummy parameters

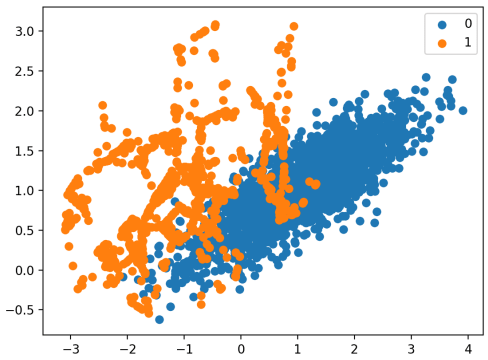


Figure 3.22: Number of classes of majority to minority class is 2:1 ratio.With x-axis = X, y-axis = Y, dummy parameters

Bibliography

[1] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, V. B. Adya, C. Affeldt, M. Agathos, K. Agatsuma, N. Aggarwal, O. D. Aguiar, L. Aiello, A. Ain, P. Ajith, B. Allen, A. Allocca, P. A. Altin, S. B. Anderson, W. G. Anderson, K. Arai, M. C. Araya, C. C. Arceneaux, J. S. Areeda, N. Arnaud, K. G. Arun, S. Ascenzi, G. Ashton, M. Ast, S. M. Aston, P. Astone, P. Aufmuth, C. Aulbert, S. Babak, P. Bacon, M. K. M. Bader, P. T. Baker, F. Baldaccini, G. Ballardin, S. W. Ballmer, J. C. Barayoga, S. E. Barclay, B. C. Barish, D. Barker, F. Barone, B. Barr, L. Barsotti, M. Barsuglia, D. Barta, J. Bartlett, I. Bartos, R. Bassiri, A. Basti, J. C. Batch, C. Baune, V. Bavigadda, M. Bazzan, B. Behnke, M. Bejger, C. Belczynski, A. S. Bell, C. J. Bell, B. K. Berger, J. Bergman, G. Bergmann, C. P. L. Berry, D. Bersanetti, A. Bertolini, J. Betzwieser, S. Bhagwat, R. Bhandare, I. A. Bilenko, G. Billingsley, J. Birch, R. Birney, S. Biscans, A. Bisht, M. Bitossi, C. Biwer, M. A. Bizouard, J. K. Blackburn, C. D. Blair, D. G. Blair, R. M. Blair, S. Bloemen, O. Bock, T. P. Bodiya, M. Boer, G. Bogaert, C. Bogan, A. Bohe, P. Bojtos, C. Bond, F. Bondu, R. Bonnand, B. A. Boom, R. Bork, V. Boschi, S. Bose, Y. Bouffanais, A. Bozzi, C. Bradaschia, P. R. Brady, V. B. Braginsky, M. Branchesi, J. E. Brau, T. Briant, A. Brillet, M. Brinkmann, V. Brisson, P. Brockill, A. F. Brooks, D. A. Brown, D. D. Brown, N. M. Brown, C. C. Buchanan, A. Buikema, T. Bulik, H. J. Bulten, A. Buonanno, D. Buskulic, C. Buy, R. L. Byer, L. Cadonati, G. Cagnoli, C. Cahillane, J. Calderón Bustillo, T. Callister, E. Calloni, J. B. Camp, K. C. Cannon, J. Cao, C. D. Capano, E. Capocasa, F. Carbognani, S. Caride, J. Casanueva Diaz, C. Casentini, S. Caudill, M. Cavaglià, F. Cavalier, R. Cavalieri, G. Cella, C. Cepeda, L. Cerboni Baiardi, G. Cerretani, E. Cesarini, R. Chakraborty, T. Chalermsongsak, S. J. Chamberlin, M. Chan, S. Chao, P. Charlton, E. Chassande-Mottin, H. Y. Chen, Y. Chen, C. Cheng, A. Chincarini, A. Chiummo, H. S. Cho, M. Cho, J. H. Chow, N. Christensen, Q. Chu, S. Chua, S. Chung, G. Ciani, F. Clara, J. A. Clark, F. Cleva, E. Coccia, P. F. Cohadon, A. Colla, C. G. Collette, L. Cominsky, J. Constancio, M., A. Conte, L. Conti, D. Cook, T. R. Corbitt, N. Cornish, A. Corsi, S. Cortese, C. A. Costa, M. W. Coughlin, S. B. Coughlin, J. P. Coulon, S. T. Countryman, P. Couvares, E. E. Cowan, D. M. Coward, M. J. Cowart, D. C. Coyne, R. Coyne, K. Craig, J. D. E. Creighton, J. Cripe, S. G. Crowder, A. Cumming, L. Cunningham, E. Cuoco, T. Dal Canton, S. L. Danilishin, S. D'Antonio, K. Danzmann, N. S. Darman, V. Dattilo, I. Dave, H. P. Daveloza, M. Davier, G. S. Davies, E. J. Daw, R. Day, D. DeBra, G. Debreczeni, J. Degallaix, M. De Laurentis, S. Deléglise, W. Del Pozzo, T. Denker, T. Dent, H. Dereli, V. Dergachev, R. DeRosa, R. T. DeRosa, R. DeSalvo, S. Dhurandhar, M. C. Díaz, L. Di Fiore, M. Di Giovanni, A. Di Lieto, S. Di Pace, I. Di Palma, A. Di Virgilio, G. Dojcinoski, V. Dolique, F. Donovan, K. L. Dooley, S. Doravari, R. Douglas, T. P. Downes, M. Drago, R. W. P. Drever, J. C. Driggers, Z. Du, M. Ducrot, S. E. Dwyer, T. B. Edo, M. C. Edwards, A. Effler,

H. B. Eggenstein, P. Ehrens, J. Eichholz, S. S. Eikenberry, W. Engels, R. C. Essick, T. Etzel, M. Evans, T. M. Evans, R. Everett, M. Factourovich, V. Fafone, H. Fair, S. Fairhurst, X. Fan, Q. Fang, S. Farinon, B. Farr, W. M. Farr, M. Favata, M. Fays, H. Fehrmann, M. M. Fejer, I. Ferrante, E. C. Ferreira, F. Ferrini, F. Fidecaro, I. Fiori, D. Fiorucci, R. P. Fisher, R. Flaminio, M. Fletcher, J. D. Fournier, S. Franco, S. Frasca, F. Frasconi, Z. Frei, A. Freise, R. Frey, V. Frey, T. T. Fricke, P. Fritschel, V. V. Frolov, P. Fulda, M. Fyffe, H. A. G. Gabbard, J. R. Gair, L. Gammaitoni, S. G. Gaonkar, F. Garufi, A. Gatto, G. Gaur, N. Gehrels, G. Gemme, B. Gendre, E. Genin, A. Gennai, J. George, L. Gergely, V. Germain, A. Ghosh, S. Ghosh, J. A. Giaime, K. D. Giardina, A. Giazotto, K. Gill, A. Glaefke, E. Goetz, R. Goetz, L. Gondan, G. González, J. M. Gonzalez Castro, A. Gopakumar, N. A. Gordon, M. L. Gorodetsky, S. E. Gossan, M. Gosselin, R. Gouaty, C. Graef, P. B. Graff, M. Granata, A. Grant, S. Gras, C. Gray, G. Greco, A. C. Green, P. Groot, H. Grote, S. Grunewald, G. M. Guidi, X. Guo, A. Gupta, M. K. Gupta, K. E. Gushwa, E. K. Gustafson, R. Gustafson, J. J. Hacker, B. R. Hall, E. D. Hall, G. Hammond, M. Haney, M. M. Hanke, J. Hanks, C. Hanna, M. D. Hannam, J. Hanson, T. Hardwick, J. Harms, G. M. Harry, I. W. Harry, M. J. Hart, M. T. Hartman, C. J. Haster, K. Haughian, A. Heidmann, M. C. Heintze, H. Heitmann, P. Hello, G. Hemming, M. Hendry, I. S. Heng, J. Hennig, A. W. Heptonstall, M. Heurs, S. Hild, D. Hoak, K. A. Hodge, D. Hofman, S. E. Hollitt, K. Holt, D. E. Holz, P. Hopkins, D. J. Hosken, J. Hough, E. A. Houston, E. J. Howell, Y. M. Hu, S. Huang, E. A. Huerta, D. Huet, B. Hughey, S. Husa, S. H. Huttner, T. Huynh-Dinh, A. Idrisy, N. Indik, D. R. Ingram, R. Inta, H. N. Isa, J. M. Isac, M. Isi, G. Islas, T. Isogai, B. R. Iyer, K. Izumi, T. Jacqmin, H. Jang, K. Jani, P. Jaranowski, S. Jawahar, F. Jiménez-Forteza, W. W. Johnson, D. I. Jones, R. Jones, R. J. G. Jonker, L. Ju, H. K, C. V. Kalaghatgi, V. Kalogera, S. Kandhasamy, G. Kang, J. B. Kanner, S. Karki, M. Kasprzack, E. Katsavounidis, W. Katzman, S. Kaufer, T. Kaur, K. Kawabe, F. Kawazoe, F. Kéfélian, M. S. Kehl, D. Keitel, D. B. Kelley, W. Kells, R. Kennedy, J. S. Key, A. Khalaidovski, F. Y. Khalili, I. Khan, S. Khan, Z. Khan, E. A. Khazanov, N. Kijbunchoo, C. Kim, J. Kim, K. Kim, N.-G. Kim, N. Kim, Y. M. Kim, E. J. King, P. J. King, D. L. Kinzel, J. S. Kissel, L. Kleybolte, S. Klimenko, S. M. Koehlenbeck, K. Kokeyama, S. Koley, V. Kondrashov, A. Kontos, M. Korobko, W. Z. Korth, I. Kowalska, D. B. Kozak, V. Kringel, B. Krishnan, A. Królak, C. Krueger, G. Kuehn, P. Kumar, L. Kuo, A. Kutynia, B. D. Lackey, M. Landry, J. Lange, B. Lantz, P. D. Lasky, A. Lazzarini, C. Lazzaro, P. Leaci, S. Leavey, E. O. Lebigot, C. H. Lee, H. K. Lee, H. M. Lee, K. Lee, A. Lenon, M. Leonardi, J. R. Leong, N. Leroy, N. Letendre, Y. Levin, B. M. Levine, T. G. F. Li, A. Libson, T. B. Littenberg, N. A. Lockerbie, J. Logue, A. L. Lombardi, J. E. Lord, M. Lorenzini, V. Loriette, M. Lormand, G. Losurdo, J. D. Lough, H. Lück, A. P. Lundgren, J. Luo, R. Lynch, Y. Ma, T. MacDonald, B. Machenschalk, M. MacInnis, D. M. Macleod, F. Magaña-Sandoval, R. M. Magee, M. Mageswaran, E. Majorana, I. Maksimovic, V. Malvezzi, N. Man, I. Mandel, V. Mandic, V. Mangano, G. L. Mansell, M. Manske, M. Mantovani, F. Marchesoni, F. Marion, S. Márka, Z. Márka, A. S. Markosyan, E. Maros, F. Martelli, L. Martellini, I. W. Martin, R. M. Martin, D. V. Martynov, J. N. Marx, K. Mason, A. Masserot, T. J. Massinger, M. Masso-Reid, F. Matichard, L. Matone, N. Mavalvala, N. Mazumder, G. Mazzolo, R. McCarthy, D. E. McClelland, S. McCormick, S. C. McGuire, G. McIntyre, J. McIver, D. J. McManus, S. T. McWilliams, D. Meacher, G. D. Meadors, J. Meidam, A. Melatos, G. Mendell, D. Mendoza-Gandara, R. A. Mercer, E. Merilh, M. Merzougui, S. Meshkov, C. Messenger, C. Messick, P. M. Meyers, F. Mezzani, H. Miao, C. Michel, H. Middleton, E. E. Mikhailov, L. Milano,

J. Miller, M. Millhouse, Y. Minenkov, J. Ming, S. Mirshekari, C. Mishra, S. Mitra, V. P. Mitrofanov, G. Mitselmakher, R. Mittleman, A. Moggi, M. Mohan, S. R. P. Mohapatra, M. Montani, B. C. Moore, C. J. Moore, D. Moraru, G. Moreno, S. R. Morriss, K. Mossavi, B. Mours, C. M. Mow-Lowry, C. L. Mueller, G. Mueller, A. W. Muir, A. Mukherjee, D. Mukherjee, S. Mukherjee, N. Mukund, A. Mullavey, J. Munch, D. J. Murphy, P. G. Murray, A. Mytidis, I. Nardecchia, L. Naticchioni, R. K. Nayak, V. Necula, K. Nedkova, G. Nelemans, M. Neri, A. Neunzert, G. Newton, T. T. Nguyen, A. B. Nielsen, S. Nissanke, A. Nitz, F. Nocera, D. Nolting, M. E. N. Normandin, L. K. Nuttall, J. Oberling, E. Ochsner, J. O'Dell, E. Oelker, G. H. Ogin, J. J. Oh, S. H. Oh, F. Ohme, M. Oliver, P. Oppermann, R. J. Oram, B. O'Reilly, R. O'Shaughnessy, D. J. Ottaway, R. S. Ottens, H. Overmier, B. J. Owen, A. Pai, S. A. Pai, J. R. Palamos, O. Palashov, C. Palomba, A. Pal-Singh, H. Pan, C. Pankow, F. Pannarale, B. C. Pant, F. Paoletti, A. Paoli, M. A. Papa, H. R. Paris, W. Parker, D. Pascucci, A. Pasqualetti, R. Passaquieti, D. Passuello, B. Patricelli, Z. Patrick, B. L. Pearlstone, M. Pedraza, R. Pedurand, L. Pekowsky, A. Pele, S. Penn, A. Perreca, M. Phelps, O. Piccinni, M. Pichot, F. Piergiovanni, V. Pierro, G. Pillant, L. Pinard, I. M. Pinto, M. Pitkin, R. Poggiani, P. Popolizio, A. Post, J. Powell, J. Prasad, V. Predoi, S. S. Premachandra, T. Prestegard, L. R. Price, M. Prijatelj, M. Principe, S. Privitera, R. Prix, G. A. Prodi, L. Prokhorov, O. Puncken, M. Punturo, P. Puppo, M. Pürrer, H. Qi, J. Qin, V. Quetschke, E. A. Quintero, R. Quitzow-James, F. J. Raab, D. S. Rabeling, H. Radkins, P. Raffai, S. Raja, M. Rakhmanov, P. Rapagnani, V. Raymond, M. Razzano, V. Re, J. Read, C. M. Reed, T. Regimbau, L. Rei, S. Reid, D. H. Reitze, H. Rew, S. D. Reyes, F. Ricci, K. Riles, N. A. Robertson, R. Robie, F. Robinet, A. Rocchi, L. Rolland, J. G. Rollins, V. J. Roma, J. D. Romano, R. Romano, G. Romanov, J. H. Romie, D. Rosińska, S. Rowan, A. Rüdiger, P. Ruggi, K. Ryan, S. Sachdev, T. Sadecki, L. Sadeghian, L. Salconi, M. Saleem, F. Salemi, A. Samajdar, L. Sammut, E. J. Sanchez, V. Sandberg, B. Sandeen, J. R. Sanders, B. Sassolas, B. S. Sathyaprakash, P. R. Saulson, O. Sauter, R. L. Savage, A. Sawadsky, P. Schale, R. Schilling, J. Schmidt, P. Schmidt, R. Schnabel, R. M. S. Schofield, A. Schönbeck, E. Schreiber, D. Schuette, B. F. Schutz, J. Scott, S. M. Scott, D. Sellers, D. Sentenac, V. Sequino, A. Sergeev, G. Serna, Y. Setyawati, A. Sevigny, D. A. Shaddock, S. Shah, M. S. Shahriar, M. Shaltev, Z. Shao, B. Shapiro, P. Shawhan, A. Sheperd, D. H. Shoemaker, D. M. Shoemaker, K. Siellez, X. Siemens, D. Sigg, A. D. Silva, D. Simakov, A. Singer, L. P. Singer, A. Singh, R. Singh, A. Singhal, A. M. Sintes, B. J. J. Slagmolen, J. R. Smith, N. D. Smith, R. J. E. Smith, E. J. Son, B. Sorazu, F. Sorrentino, T. Souradeep, A. K. Srivastava, A. Staley, M. Steinke, J. Steinlechner, S. Steinlechner, D. Steinmeyer, B. C. Stephens, S. P. Stevenson, R. Stone, K. A. Strain, N. Straniero, G. Stratta, N. A. Strauss, S. Strigin, R. Sturani, A. L. Stuver, T. Z. Summerscales, L. Sun, P. J. Sutton, B. L. Swinkels, M. J. Szczepańczyk, M. Tacca, D. Talukder, D. B. Tanner, M. Tápai, S. P. Tarabrin, A. Taracchini, R. Taylor, T. Theeg, M. P. Thirugnanasambandam, E. G. Thomas, M. Thomas, P. Thomas, K. A. Thorne, K. S. Thorne, E. Thrane, S. Tiwari, V. Tiwari, K. V. Tokmakov, C. Tomlinson, M. Tonelli, C. V. Torres, C. I. Torrie, D. Töyrä, F. Travasso, G. Traylor, D. Trifirò, M. C. Tringali, L. Trozzo, M. Tse, M. Turconi, D. Tuyenbayev, D. Ugolini, C. S. Unnikrishnan, A. L. Urban, S. A. Usman, H. Vahlbruch, G. Vajente, G. Valdes, N. van Bakel, M. van Beuzekom, J. F. J. van den Brand, C. van den Broeck, D. C. Vander-Hyde, L. van der Schaaf, J. V. van Heijningen, A. A. van Veggel, M. Vardaro, S. Vass, M. Vasúth, R. Vaulin, A. Vecchio, G. Vedovato, J. Veitch, P. J. Veitch, K. Venkateswara, D. Verkindt, F. Vetrano, A. Viceré, S. Vinciguerra, D. J. Vine,

J. Y. Vinet, S. Vitale, T. Vo, H. Vocca, C. Vorvick, D. Voss, W. D. Vousden, S. P. Vyatchanin, A. R. Wade, L. E. Wade, M. Wade, M. Walker, L. Wallace, S. Walsh, G. Wang, H. Wang, M. Wang, X. Wang, Y. Wang, R. L. Ward, J. Warner, M. Was, B. Weaver, L. W. Wei, M. Weinert, A. J. Weinstein, R. Weiss, T. Welborn, L. Wen, P. Weßels, T. Westphal, K. Wette, J. T. Whelan, D. J. White, B. F. Whiting, R. D. Williams, A. R. Williamson, J. L. Willis, B. Willke, M. H. Wimmer, W. Winkler, C. C. Wipf, H. Wittel, G. Woan, J. Worden, J. L. Wright, G. Wu, J. Yablon, W. Yam, H. Yamamoto, C. C. Yancey, M. J. Yap, H. Yu, M. Yvert, A. Zadrożny, L. Zangrando, M. Zanolin, J. P. Zendri, M. Zevin, F. Zhang, L. Zhang, M. Zhang, Y. Zhang, C. Zhao, M. Zhou, Z. Zhou, X. J. Zhu, M. E. Zucker, S. E. Zuraw, and, J. Zweizig, LIGO Scientific Collaboration, and Virgo Collaboration. Astrophysical Implications of the Binary Black-hole Merger GW150914. , 818(2):L22, Feb. 2016. doi: 10.3847/2041-8205/818/2/L22.

[2] R. Abbott, T. Abbott, S. Abraham, F. Acernese, K. Ackley, A. Adams, C. Adams, R. Adhikari, V. Adya, C. Affeldt, D. Agarwal, M. Agathos, K. Agatsuma, N. Aggarwal, O. Aguiar, L. Aiello, A. Ain, P. Ajith, T. Akutsu, K. Aleman, G. Allen, A. Allocca, P. Altin, A. Amato, S. Anand, A. Ananyeva, S. Anderson, W. Anderson, M. Ando, S. Angelova, S. Ansoldi, J. Antelis, S. Antier, S. Appert, K. Arai, K. Arai, Y. Arai, S. Araki, A. Araya, M. Araya, J. Areeda, M. Arène, N. Aritomi, N. Arnaud, S. Aronson, H. Asada, Y. Asali, G. Ashton, Y. Aso, S. Aston, P. Astone, F. Aubin, P. Aufmuth, K. AultONeal, C. Austin, S. Babak, F. Badaracco, M. Bader, S. Bae, Y. Bae, A. Baer, S. Bagnasco, Y. Bai, L. Baiotti, J. Baird, R. Bajpai, M. Ball, G. Ballardin, S. Ballmer, M. Bals, A. Balsamo, G. Baltus, S. Banagiri, D. Bankar, R. Bankar, J. Barayoga, C. Barbieri, B. Barish, D. Barker, P. Barneo, F. Barone, B. Barr, L. Barsotti, M. Barsuglia, D. Barta, J. Bartlett, M. Barton, I. Bartos, R. Bassiri, A. Basti, M. Bawaj, J. Bayley, A. Baylor, M. Bazzan, B. Bécsy, V. Bedakihale, M. Bejger, I. Belahcene, V. Benedetto, D. Beniwal, M. Benjamin, T. Bennett, J. Bentley, M. BenYaala, F. Bergamin, B. Berger, S. Bernuzzi, D. Bersanetti, A. Bertolini, J. Betzwieser, R. Bhandare, A. Bhandari, D. Bhattacharjee, S. Bhaumik, J. Bidler, I. Bilenko, G. Billingsley, R. Birney, O. Birnholtz, S. Biscans, M. Bischi, S. Biscoveanu, A. Bisht, B. Biswas, M. Bitossi, M.-A. Bizouard, J. Blackburn, J. Blackman, C. Blair, D. Blair, R. Blair, F. Bobba, N. Bode, M. Boer, G. Bogaert, M. Boldrini, F. Bondu, E. Bonilla, R. Bonnand, P. Booker, B. Boom, R. Bork, V. Boschi, N. Bose, S. Bose, V. Bossilkov, V. Boudart, Y. Bouffanais, A. Bozzi, C. Bradaschia, P. Brady, A. Bramley, A. Branch, M. Branchesi, J. Brau, M. Breschi, T. Briant, J. Briggs, A. Brillet, M. Brinkmann, P. Brockill, A. Brooks, J. Brooks, D. Brown, S. Brunett, G. Bruno, R. Bruntz, J. Bryant, A. Buikema, T. Bulik, H. Bulten, A. Buonanno, R. Buscicchio, D. Buskulic, R. Byer, L. Cadonati, M. Caesar, G. Cagnoli, C. Cahillane, H. Cain, J. C. Bustillo, J. Callaghan, T. Callister, E. Calloni, J. Camp, M. Canepa, M. Cannavacciuolo, K. Cannon, H. Cao, J. Cao, Z. Cao, E. Capocasa, E. Capote, G. Carapella, F. Carbognani, J. Carlin, M. Carney, M. Carpinelli, G. Carullo, T. Carver, J. C. Diaz, C. Casentini, G. Castaldi, S. Caudill, M. Cavaglià, F. Cavalier, R. Cavalieri, G. Cella, P. Cerdá-Durán, E. Cesarini, W. Chaibi, K. Chakravarti, B. Champion, C.-H. Chan, C. Chan, C. Chan, M. Chan, K. Chandra, P. Chanial, S. Chao, P. Charlton, E. Chase, E. Chassande-Mottin, D. Chatterjee, M. Chaturvedi, A. Chen, C. Chen, H. Chen, J. Chen, K. Chen, X. Chen, Y.-B. Chen, Y.-R. Chen, Z. Chen, H. Cheng, C. Cheong, H. Cheung, H. Chia, F. Chiadini, C.-Y. Chiang, R. Chierici, A. Chincarini, M. Chiofalo, A. Chiummo, G. Cho, H. Cho, S. Choate, R. Choudhary, S. Choudhary, N. Christensen, H. Chu, Q. Chu, Y.-K. Chu, S. Chua, K. Chung, G. Ciani, P. Ciecielag, M. Cieślar,

M. Cifaldi, A. Ciobanu, R. Ciolfi, F. Cipriano, A. Cirone, F. Clara, E. Clark, J. Clark, L. Clarke, P. Clearwater, S. Clesse, F. Cleva, E. Coccia, P.-F. Cohadon, D. Cohen, L. Cohen, M. Colleoni, C. Collette, M. Colpi, C. Compton, M. Constancio, L. Conti, S. Cooper, P. Corban, T. Corbitt, I. Cordero-Carrión, S. Corezzi, K. Corley, N. Cornish, D. Corre, A. Corsi, S. Cortese, C. Costa, R. Cotesta, M. Coughlin, S. Coughlin, J.-P. Coulon, S. Countryman, B. Cousins, P. Couvares, P. Covas, D. Coward, M. Cowart, D. Coyne, R. Coyne, J. Creighton, T. Creighton, A. Criswell, M. Croquette, S. Crowder, J. Cudell, T. Cullen, A. Cumming, R. Cummings, E. Cuoco, M. Curyło, T. D. Canton, G. Dálya, A. Dana, L. DaneshgaranBajastani, B. D'Angelo, S. Danilishin, S. D'Antonio, K. Danzmann, C. Darsow-Fromm, A. Dasgupta, L. Datrier, V. Dattilo, I. Dave, M. Davier, G. Davies, D. Davis, E. Daw, R. Dean, D. DeBra, M. Deenadayalan, J. Degallaix, M. D. Laurentis, S. Deléglise, V. D. Favero, F. D. Lillo, N. D. Lillo, W. D. Pozzo, L. DeMarchi, F. D. Matteis, V. D'Emilio, N. Demos, T. Dent, A. Depasse, R. D. Pietri, R. D. Rosa, C. D. Rossi, R. DeSalvo, R. D. Simone, S. Dhurandhar, M. Díaz, M. Diaz-Ortiz, N. Didio, T. Dietrich, L. D. Fiore, C. D. Fronzo, C. D. Giorgio, F. D. Giovanni, T. D. Girolamo, A. D. Lieto, B. Ding, S. D. Pace, I. D. Palma, F. D. Renzo, A. Divakarla, A. Dmitriev, Z. Doctor, L. D'Onofrio, F. Donovan, K. Dooley, S. Doravari, I. Dorrington, M. Drago, J. Driggers, Y. Drori, Z. Du, J.-G. Ducoin, P. Dupej, O. Durante, D. D'Urso, P.-A. Duverne, S. Dwyer, P. Easter, M. Ebersold, G. Eddolls, B. Edelman, T. Edo, O. Edy, A. Effler, S. Eguchi, J. Eichholz, S. Eikenberry, M. Eisenmann, R. Eisenstein, A. Ejlli, Y. Enomoto, L. Errico, R. Essick, H. Estellés, D. Estevez, Z. Etienne, T. Etzel, M. Evans, T. Evans, B. Ewing, V. Fafone, H. Fair, S. Fairhurst, X. Fan, A. Farah, S. Farinon, B. Farr, W. Farr, N. Farrow, E. Fauchon-Jones, M. Favata, M. Fays, M. Fazio, J. Feicht, M. Fejer, F. Feng, E. Fenyvesi, D. Ferguson, A. Fernandez-Galiana, I. Ferrante, T. Ferreira, F. Fidecaro, P. Figura, I. Fiori, M. Fishbach, R. Fisher, R. Fittipaldi, V. Fiumara, R. Flaminio, E. Floden, E. Flynn, H. Fong, J. Font, B. Fornal, P. Forsyth, A. Franke, S. Frasca, F. Frasconi, C. Frederick, Z. Frei, A. Freise, R. Frey, P. Fritschel, V. Frolov, G. Fronzé, Y. Fujii, Y. Fujikawa, M. Fukunaga, M. Fukushima, P. Fulda, M. Fyffe, H. Gabbard, B. Gadre, S. Gaebel, J. Gair, J. Gais, S. Galaudage, R. Gamba, D. Ganapathy, A. Ganguly, D. Gao, S. Gaonkar, B. Garaventa, C. García-Núñez, C. García-Quirós, F. Garufi, B. Gateley, S. Gaudio, V. Gayathri, G. Ge, G. Gemme, A. Gennai, J. George, L. Gergely, P. Gewecke, S. Ghonge, A. Ghosh, A. Ghosh, S. Ghosh, S. Ghosh, S. Ghosh, B. Giacomazzo, L. Giacoppo, J. Giaime, K. Giardina, D. Gibson, C. Gier, M. Giesler, P. Giri, F. Gissi, J. Glanzer, A. Gleckl, P. Godwin, E. Goetz, R. Goetz, N. Gohlke, B. Goncharov, G. González, A. Gopakumar, M. Gosselin, R. Gouaty, B. Grace, A. Grado, M. Granata, V. Granata, A. Grant, S. Gras, P. Grassia, C. Gray, R. Gray, G. Greco, A. Green, R. Green, A. Gretarsson, E. Gretarsson, D. Griffith, W. Griffiths, H. Griggs, G. Grignani, A. Grimaldi, E. Grimes, S. Grimm, H. Grote, S. Grunewald, P. Gruning, J. Guerrero, G. Guidi, A. Guimaraes, G. Guixé, H. Gulati, H.-K. Guo, Y. Guo, A. Gupta, A. Gupta, P. Gupta, E. Gustafson, R. Gustafson, F. Guzman, S. Ha, L. Haegel, A. Hagiwara, S. Haino, O. Halim, E. Hall, E. Hamilton, G. Hammond, W.-B. Han, M. Haney, J. Hanks, C. Hanna, M. Hannam, O. Hannuksela, H. Hansen, T. Hansen, J. Hanson, T. Harder, T. Hardwick, K. Haris, J. Harms, G. Harry, I. Harry, D. Hartwig, K. Hasegawa, B. Haskell, R. Hasskew, C.-J. Haster, K. Hattori, K. Haughian, H. Hayakawa, K. Hayama, F. Hayes, J. Healy, A. Heidmann, M. Heintze, J. Heinze, J. Heinzel, H. Heitmann, F. Hellman, P. Hello, A. Helmling-Cornell, G. Hemming, M. Hendry, I. Heng, E. Hennes, J. Hennig, M. Hennig, F. H. Vivanco, M. Heurs, S. Hild, P. Hill, Y. Himemoto, A. Hines, Y. Hi-

ranuma, N. Hirata, E. Hirose, S. Hochheim, D. Hofman, J. Hohmann, A. Holgado, N. Holland, I. Hollows, Z. Holmes, K. Holt, D. Holz, Z. Hong, P. Hopkins, J. Hough, E. Howell, C. Hoy, D. Hoyland, A. Hreibi, B.-H. Hsieh, Y. Hsu, G.-Z. Huang, H.-Y. Huang, P. Huang, Y.-C. Huang, Y.-J. Huang, Y.-W. Huang, M. Hübner, A. Huddart, E. Huerta, B. Hughey, D. Hui, V. Hui, S. Husa, S. Huttner, R. Huxford, T. Huynh-Dinh, S. Ide, B. Idzkowski, A. Iess, B. Ikenoue, S. Imam, K. Inayoshi, H. Inchauspe, C. Ingram, Y. Inoue, G. Intini, K. Ioka, M. Isi, K. Isleif, K. Ito, Y. Itoh, B. Iyer, K. Izumi, V. JaberianHamedan, T. Jacqmin, S. Jadhav, S. Jadhav, A. James, A. Jan, K. Jani, K. Janssens, N. Janthalur, P. Jaranowski, D. Jariwala, R. Jaume, A. Jenkins, C. Jeon, M. Jeunon, W. Jia, J. Jiang, H.-B. Jin, G. Johns, A. Jones, D. Jones, J. Jones, P. Jones, R. Jones, R. Jonker, L. Ju, K. Jung, P. Jung, J. Junker, K. Kaihotsu, T. Kajita, M. Kakizaki, C. Kalaghatgi, V. Kalogera, B. Kamai, M. Kamiizumi, N. Kanda, S. Kandhasamy, G. Kang, J. Kanner, Y. Kao, S. Kapadia, D. Kapasi, S. Karat, C. Karathanasis, S. Karki, R. Kashyap, M. Kasprzack, W. Kastaun, S. Katsanevas, E. Katsavounidis, W. Katzman, T. Kaur, K. Kawabe, K. Kawaguchi, N. Kawai, T. Kawasaki, F. Kéfélian, D. Keitel, J. Key, S. Khadka, F. Khalili, I. Khan, S. Khan, E. Khazanov, N. Khetan, M. Khursheed, N. Kijbunchoo, C. Kim, J. Kim, J. Kim, K. Kim, W. Kim, Y.-M. Kim, C. Kimball, N. Kimura, P. King, M. Kinley-Hanlon, R. Kirchhoff, J. Kissel, N. Kita, H. Kitazawa, L. Kleybolte, S. Klimenko, A. Knee, T. Knowles, E. Knyazev, P. Koch, G. Koekoek, Y. Kojima, K. Kokeyama, S. Koley, P. Kolitsidou, M. Kolstein, K. Komori, V. Kondrashov, A. Kong, A. Kontos, N. Koper, M. Korobko, K. Kotake, M. Kovalam, D. Kozak, C. Kozakai, R. Kozu, V. Kringel, N. Krishnendu, A. Królak, G. Kuehn, F. Kuei, A. Kumar, P. Kumar, R. Kumar, R. Kumar, J. Kume, K. Kuns, C. Kuo, H.-S. Kuo, Y. Kuromiya, S. Kuroyanagi, K. Kusayanagi, K. Kwak, S. Kwang, D. Laghi, E. Lalande, T. Lam, A. Lamberts, M. Landry, B. Lane, R. Lang, J. Lange, B. Lantz, I. L. Rosa, A. Lartaux-Vollard, P. Lasky, M. Laxen, A. Lazzarini, C. Lazzaro, P. Leaci, S. Leavey, Y. Lecoeuche, H. Lee, H. Lee, H. Lee, J. Lee, K. Lee, R. Lee, J. Lehmann, A. Lemaître, E. Leon, M. Leonardi, N. Leroy, N. Letendre, Y. Levin, J. Leviton, A. Li, B. Li, J. Li, K. Li, T. Li, X. Li, C.-Y. Lin, F.-K. Lin, F.-L. Lin, H. Lin, L.-C. Lin, F. Linde, S. Linker, J. Linley, T. Littenberg, G. Liu, J. Liu, K. Liu, X. Liu, M. Llorens-Monteagudo, R. Lo, A. Lockwood, M. Lollie, L. London, A. Longo, D. Lopez, M. Lorenzini, V. Loriette, M. Lormand, G. Losurdo, J. Lough, C. Lousto, G. Lovelace, H. Lück, D. Lumaca, A. Lundgren, L.-W. Luo, R. Macas, M. MacInnis, D. Macleod, I. MacMillan, A. Macquet, I. M. Hernandez, F. Magaña-Sandoval, C. Magazzù, R. Magee, R. Maggiore, E. Majorana, C. Makarem, I. Maksimovic, S. Maliakal, A. Malik, N. Man, V. Mandic, V. Mangano, J. Mango, G. Mansell, M. Manske, M. Mantovani, M. Mapelli, F. Marchesoni, M. Marchio, F. Marion, Z. Mark, S. Márka, Z. Márka, C. Markakis, A. Markosyan, A. Markowitz, E. Maros, A. Marquina, S. Marsat, F. Martelli, I. Martin, R. Martin, M. Martinez, V. Martinez, K. Martinovic, D. Martynov, E. Marx, H. Masalehdan, K. Mason, E. Massera, A. Masserot, T. Massinger, M. Masso-Reid, S. Mastrogiovanni, A. Matas, M. Mateu-Lucena, F. Matichard, M. Matiushechkina, N. Mavalvala, J. McCann, R. McCarthy, D. McClelland, P. McClincy, S. McCormick, L. McCuller, G. McGhee, S. McGuire, C. McIsaac, J. McIver, D. McManus, T. McRae, S. McWilliams, D. Meacher, M. Mehmet, A. Mehta, A. Melatos, D. Melchor, G. Mendell, A. Menendez-Vazquez, C. Menoni, R. Mercer, L. Mereni, K. Merfeld, E. Merilh, J. Merritt, M. Merzougui, S. Meshkov, C. Messenger, C. Messick, P. Meyers, F. Meylahn, A. Mhaske, A. Miani, H. Miao, I. Michaloliakos, C. Michel, Y. Michimura, H. Middleton, L. Milano, A. Miller, M. Millhouse, J. Mills, E. Milotti, M. Milovich-

Goff, O. Minazzoli, Y. Minenkov, N. Mio, L. Mir, A. Mishkin, C. Mishra, T. Mishra, T. Mistry, S. Mitra, V. Mitrofanov, G. Mitselmakher, R. Mittleman, O. Miyakawa, A. Miyamoto, Y. Miyazaki, K. Miyo, S. Miyoki, G. Mo, K. Mogushi, S. Mohapatra, S. Mohite, I. Molina, M. Molina-Ruiz, M. Mondin, M. Montani, C. Moore, D. Moraru, F. Morawski, A. More, C. Moreno, G. Moreno, Y. Mori, S. Morisaki, Y. Moriwaki, B. Mours, C. Mow-Lowry, S. Mozzon, F. Muciaccia, A. Mukherjee, D. Mukherjee, S. Mukherjee, S. Mukherjee, N. Mukund, A. Mullavey, J. Munch, E. Muñiz, P. Murray, R. Musenich, S. Nadji, K. Nagano, S. Nagano, K. Nakamura, H. Nakano, M. Nakano, R. Nakashima, Y. Nakayama, I. Nardecchia, T. Narikawa, L. Naticchioni, B. Nayak, R. Nayak, R. Negishi, B. Neil, J. Neilson, G. Nelemans, T. Nelson, M. Nery, A. Neunzert, K. Ng, S. Ng, C. Nguyen, P. Nguyen, T. Nguyen, L. N. Quynh, W.-T. Ni, S. Nichols, A. Nishizawa, S. Nissanke, F. Nocera, M. Noh, M. Norman, C. North, S. Nozaki, L. Nuttall, J. Oberling, B. O'Brien, Y. Obuchi, J. O'Dell, W. Ogaki, G. Oganesyan, J. Oh, K. Oh, S. Oh, M. Ohashi, N. Ohishi, M. Ohkawa, F. Ohme, H. Ohta, M. Okada, Y. Okutani, K. Okutomi, C. Olivetto, K. Oohara, C. Ooi, R. Oram, B. O'Reilly, R. Ormiston, N. Ormsby, L. Ortega, R. O'Shaughnessy, E. O'Shea, S. Oshino, S. Ossokine, C. Osthelder, S. Otabe, D. Ottaway, H. Overmier, A. Pace, G. Pagano, M. Page, G. Pagliaroli, A. Pai, S. Pai, J. Palamos, O. Palashov, C. Palomba, K. Pan, P. Panda, H. Pang, P. Pang, C. Pankow, F. Pannarale, B. Pant, F. Paoletti, A. Paoli, A. Paolone, A. Parisi, J. Park, W. Parker, D. Pascucci, A. Pasqualetti, R. Passaquieti, D. Passuello, M. Patel, B. Patricelli, E. Payne, T. Pechsiri, M. Pedraza, M. Pegoraro, A. Pele, F. P. Arellano, S. Penn, A. Perego, A. Pereira, T. Pereira, C. Perez, C. Périgois, A. Perreca, S. Perriès, J. Petermann, D. Petterson, H. Pfeiffer, K. Pham, K. Phukon, O. Piccinni, M. Pichot, M. Piendibene, F. Piergiovanni, L. Pierini, V. Pierro, G. Pillant, F. Pilo, L. Pinard, I. Pinto, B. Piotrzkowski, K. Piotrzkowski, M. Pirello, M. Pitkin, E. Placidi, W. Plastino, C. Pluchar, R. Poggiani, E. Polini, D. Pong, S. Ponrathnam, P. Popolizio, E. Porter, J. Powell, M. Pracchia, T. Pradier, A. Prajapati, K. Prasai, R. Prasanna, G. Pratten, T. Prestegard, M. Principe, G. Prodi, L. Prokhorov, P. Prosposito, L. Prudenzi, A. Puecher, M. Punturo, F. Puosi, P. Puppo, M. Pürrer, H. Qi, V. Quetschke, P. Quinonez, R. Quitzow-James, F. Raab, G. Raaijmakers, H. Radkins, N. Radulesco, P. Raffai, S. Rail, S. Raja, C. Rajan, K. Ramirez, T. Ramirez, A. Ramos-Buades, J. Rana, P. Rapagnani, U. Rapol, B. Ratto, V. Raymond, N. Raza, M. Razzano, J. Read, L. Rees, T. Regimbau, L. Rei, S. Reid, D. Reitze, P. Relton, A. Renzini, P. Rettegno, F. Ricci, C. Richardson, J. Richardson, L. Richardson, P. Ricker, G. Riemenschneider, K. Riles, M. Rizzo, N. Robertson, R. Robie, F. Robinet, A. Rocchi, J. Rocha, S. Rodriguez, R. Rodriguez-Soto, L. Rolland, J. Rollins, V. Roma, M. Romanelli, J. Romano, R. Romano, C. Romel, A. Romero, I. Romero-Shaw, J. Romie, C. Rose, D. Rosińska, S. Rosofsky, M. Ross, S. Rowan, S. Rowlinson, S. Roy, S. Roy, D. Rozza, P. Ruggi, K. Ryan, S. Sachdev, T. Sadecki, J. Sadiq, N. Sago, S. Saito, Y. Saito, K. Sakai, Y. Sakai, M. Sakellariadou, Y. Sakuno, O. Salafia, L. Salconi, M. Saleem, F. Salemi, A. Samajdar, E. Sanchez, J. Sanchez, L. Sanchez, N. Sanchis-Gual, J. Sanders, A. Sanuy, T. Saravanan, N. Sarin, B. Sassolas, H. Satari, S. Sato, T. Sato, O. Sauter, R. Savage, V. Savant, T. Sawada, D. Sawant, H. Sawant, S. Sayah, D. Schaetzl, M. Scheel, J. Scheuer, A. Schindler-Tyka, P. Schmidt, R. Schnabel, M. Schneewind, R. Schofield, A. Schönbeck, B. Schulte, B. Schutz, E. Schwartz, J. Scott, S. Scott, M. Seglar-Arroyo, E. Seidel, T. Sekiguchi, Y. Sekiguchi, D. Sellers, A. Sergeev, A. Sengupta, N. Sennett, D. Sentenac, E. Seo, V. Sequino, Y. Setyawati, T. Shaffer, M. Shahriar, B. Shams, L. Shao, S. Sharifi, A. Sharma, P. Sharma, P. Shawhan, N. Shcheblanov, H. Shen, S. Shibagaki, M. Shikauchi, R. Shimizu, T. Shimoda, K. Shi-

mode, R. Shink, H. Shinkai, T. Shishido, A. Shoda, D. Shoemaker, D. Shoemaker, K. Shukla, S. ShyamSundar, M. Sieniawska, D. Sigg, L. Singer, D. Singh, N. Singh, A. Singha, A. Sintes, V. Sipala, V. Skliris, B. Slagmolen, T. Slaven-Blair, J. Smetana, J. Smith, R. Smith, S. Somala, K. Somiya, E. Son, K. Soni, S. Soni, B. Sorazu, V. Sordini, F. Sorrentino, N. Sorrentino, H. Sotani, R. Soulard, T. Souradeep, E. Sowell, V. Spagnuolo, A. Spencer, M. Spera, A. Srivastava, V. Srivastava, K. Staats, C. Stachie, D. Steer, J. Steinlechner, S. Steinlechner, D. Stops, M. Stover, K. Strain, L. Strang, G. Stratta, A. Strunk, R. Sturani, A. Stuver, J. Südbeck, S. Sudhagar, V. Sudhir, R. Sugimoto, H. Suh, T. Summerscales, H. Sun, L. Sun, S. Sunil, A. Sur, J. Suresh, P. Sutton, T. Suzuki, T. Suzuki, B. Swinkels, M. Szczepańczyk, P. Szewczyk, M. Tacca, H. Tagoshi, S. Tait, H. Takahashi, R. Takahashi, A. Takamori, S. Takano, H. Takeda, M. Takeda, C. Talbot, H. Tanaka, K. Tanaka, K. Tanaka, T. Tanaka, T. Tanaka, A. Tanasijczuk, S. Tanioka, D. Tanner, D. Tao, A. Tapia, E. T. S. Martin, E. T. S. Martin, J. Tasson, S. Telada, R. Tenorio, L. Terkowski, M. Test, M. Thirugnanasambandam, M. Thomas, P. Thomas, J. Thompson, S. Thondapu, K. Thorne, E. Thrane, S. Tiwari, S. Tiwari, V. Tiwari, K. Toland, A. Tolley, T. Tomaru, Y. Tomigami, T. Tomura, M. Tonelli, A. Torres-Forné, C. Torrie, I. T. e Melo, D. Töyrä, A. Trapananti, F. Travasso, G. Traylor, M. Tringali, A. Tripathee, L. Troiano, A. Trovato, L. Trozzo, R. Trudeau, D. Tsai, D. Tsai, K. Tsang, T. Tsang, J.-S. Tsao, M. Tse, R. Tso, K. Tsubono, S. Tsuchida, L. Tsukada, D. Tsuna, T. Tsutsui, T. Tsuzuki, M. Turconi, D. Tuyenbayev, A. Ubhi, N. Uchikata, T. Uchiyama, R. Udall, A. Ueda, T. Uehara, K. Ueno, G. Ueshima, D. Ugolini, C. Unnikrishnan, F. Uraguchi, A. Urban, T. Ushiba, S. Usman, A. Utina, H. Vahlbruch, G. Vajente, A. Vajpeyi, G. Valdes, M. Valentini, V. Valsan, N. van Bakel, M. van Beuzekom, J. van den Brand, C. V. D. Broeck, D. Vander-Hyde, L. van der Schaaf, J. van Heijningen, J. Vanosky, M. van Putten, N. van Remortel, M. Vardaro, A. Vargas, V. Varma, M. Vasúth, A. Vecchio, G. Vedovato, J. Veitch, P. Veitch, K. Venkateswara, J. Venneberg, G. Venugopalan, D. Verkindt, Y. Verma, D. Veske, F. Vetrano, A. Viceré, A. Viets, V. Villa-Ortega, J.-Y. Vinet, S. Vitale, T. Vo, H. Vocca, E. von Reis, J. von Wrangel, C. Vorvick, S. Vyatchanin, L. Wade, M. Wade, K. Wagner, R. Walet, M. Walker, G. Wallace, L. Wallace, S. Walsh, J. Wang, J. Wang, W. Wang, R. Ward, J. Warner, M. Was, T. Washimi, N. Washington, J. Watchi, B. Weaver, L. Wei, M. Weinert, A. Weinstein, R. Weiss, C. Weller, F. Wellmann, L. Wen, P. Weßels, J. Westhouse, K. Wette, J. Whelan, D. White, B. Whiting, C. Whittle, D. Wilken, D. Williams, M. Williams, A. Williamson, J. Willis, B. Willke, D. Wilson, W. Winkler, C. Wipf, T. Wlodarczyk, G. Woan, J. Woehler, J. Wofford, I. Wong, C. Wu, D. Wu, H. Wu, S. Wu, D. Wysocki, L. Xiao, W.-R. Xu, T. Yamada, H. Yamamoto, K. Yamamoto, K. Yamamoto, T. Yamamoto, K. Yamashita, R. Yamazaki, F. Yang, L. Yang, Y. Yang, Y. Yang, Z. Yang, M. Yap, D. Yeeles, A. Yelikar, M. Ying, K. Yokogawa, J. Yokoyama, T. Yokozawa, A. Yoon, T. Yoshioka, H. Yu, H. Yu, H. Yuzurihara, A. Zadrożny, M. Zanolin, S. Zeidler, T. Zelenova, J.-P. Zendri, M. Zevin, M. Zhan, H. Zhang, J. Zhang, L. Zhang, R. Zhang, T. Zhang, C. Zhao, G. Zhao, Y. Zhao, Y. Zhao, Z. Zhou, X. Zhu, Z.-H. Zhu, A. Zimmerman, M. Zucker, and J. Z. and. Search for anisotropic gravitational-wave backgrounds using data from advanced LIGO and advanced virgo's first three observing runs. *Physical Review D*, 104(2), jul 2021. doi: 10.1103/physrevd.104.022005. URL https://doi.org/10.1103%2Fphysrevd.104.022005.

[3] F. Acernese, M. Agathos, K. Agatsuma, D. Aisa, N. Allemandou, A. Allocca, J. Amarni, P. Astone, G. Balestri, G. Ballardin, F. Barone, J.-P. Baronick, M. Barsuglia, A. Basti, F. Basti, T. S. Bauer, V. Bavigadda, M. Bejger, M. G. Beker, C. Bel-

czynski, D. Bersanetti, A. Bertolini, M. Bitossi, M. A. Bizouard, S. Bloemen, M. Blom, M. Boer, G. Bogaert, D. Bondi, F. Bondu, L. Bonelli, R. Bonnand, V. Boschi, L. Bosi, T. Bouedo, C. Bradaschia, M. Branchesi, T. Briant, A. Brillet, V. Brisson, T. Bulik, H. J. Bulten, D. Buskulic, C. Buy, G. Cagnoli, E. Calloni, C. Campeggi, B. Canuel, F. Carbognani, F. Cavalier, R. Cavalieri, G. Cella, E. Cesarini, E. Chassande-Mottin, A. Chincarini, A. Chiummo, S. Chua, F. Cleva, E. Coccia, P.-F. Cohadon, A. Colla, M. Colombini, A. Conte, J.-P. Coulon, E. Cuoco, A. Dalmaz, S. D'Antonio, V. Dattilo, M. Davier, R. Day, G. Debreczeni, J. Degallaix, S. Delé glise, W. D. Pozzo, H. Dereli, R. D. Rosa, L. D. Fiore, A. D. Lieto, A. D. Virgilio, M. Doets, V. Dolique, M. Drago, M. Ducrot, G. Endrőczi, V. Fafone, S. Farinon, I. Ferrante, F. Ferrini, F. Fidecaro, I. Fiori, R. Flaminio, J.-D. Fournier, S. Franco, S. Frasca, F. Frasconi, L. Gammaitoni, F. Garufi, M. Gaspard, A. Gatto, G. Gemme, B. Gendre, E. Genin, A. Gennai, S. Ghosh, L. Giacobone, A. Giazotto, R. Gouaty, M. Granata, G. Greco, P. Groot, G. M. Guidi, J. Harms, A. Heidmann, H. Heitmann, P. Hello, G. Hemming, E. Hennes, D. Hofman, P. Jaranowski, R. J. G. Jonker, M. Kasprzack, F. Kéfélian, I. Kowalska, M. Kraan, A. Królak, A. Kutynia, C. Lazzaro, M. Leonardi, N. Leroy, N. Letendre, T. G. F. Li, B. Lieunard, M. Lorenzini, V. Loriette, G. Losurdo, C. Magazzù, E. Majorana, I. Maksimovic, V. Malvezzi, N. Man, V. Mangano, M. Mantovani, F. Marchesoni, F. Marion, J. Marque, F. Martelli, L. Martellini, A. Masserot, D. Meacher, J. Meidam, F. Mezzani, C. Michel, L. Milano, Y. Minenkov, A. Moggi, M. Mohan, M. Montani, N. Morgado, B. Mours, F. Mul, M. F. Nagy, I. Nardecchia, L. Naticchioni, G. Nelemans, I. Neri, M. Neri, F. Nocera, E. Pacaud, C. Palomba, F. Paoletti, A. Paoli, A. Pasqualetti, R. Passaquieti, D. Passuello, M. Perciballi, S. Petit, M. Pichot, F. Piergiovanni, G. Pillant, A. Piluso, L. Pinard, R. Poggiani, M. Prijatelj, G. A. Prodi, M. Punturo, P. Puppo, D. S. Rabeling, I. Rácz, P. Rapagnani, M. Razzano, V. Re, T. Regimbau, F. Ricci, F. Robinet, A. Rocchi, L. Rolland, R. Romano, D. Rosińska, P. Ruggi, E. Saracco, B. Sassolas, F. Schimmel, D. Sentenac, V. Sequino, S. Shah, K. Siellez, N. Straniero, B. Swinkels, M. Tacca, M. Tonelli, F. Travasso, M. Turconi, G. Vajente, N. van Bakel, M. van Beuzekom, J. F. J. van den Brand, C. V. D. Broeck, M. V. van der Sluys, J. van Heijningen, M. Vasúth, G. Vedovato, J. Veitch, D. Verkindt, F. Vetrano, A. Viceré, J.-Y. Vinet, G. Visser, H. Vocca, R. Ward, M. Was, L.-W. Wei, M. Yvert, A. Z. żny, and J.-P. Zendri. Advanced virgo: a second-generation interferometric gravitational wave detector. *Classical and Quantum Gravity*, 32(2):024001, dec 2014. doi: 10.1088/0264-9381/32/2/024001. URL https://doi.org/10.1088%2F0264-9381%2F32%2F2%2F024001.

[4] H. Ge, R. F. Webbink, X. Chen, and Z. Han. Adiabatic Mass Loss in Binary Stars. III. From the Base of the Red Giant Branch to the Tip of the Asymptotic Giant Branch. , 899(2):132, Aug. 2020. doi: 10.3847/1538-4357/aba7b7.

[5] H. Ge, R. F. Webbink, and Z. Han. The Thermal Equilibrium Mass-loss Model and Its Applications in Binary Evolution. , 249(1):9, July 2020. doi: 10.3847/1538-4365/ab98f6.

[6] J. R. Hurley, O. R. Pols, and C. A. Tout. Comprehensive analytic formulae for stellar evolution as a function of mass and metallicity. *Monthly Notices of the Royal Astronomical Society*, 315(3):543–569, 07 2000. ISSN 0035-8711. doi: 10.1046/j.1365-8711.2000.03426.x. URL https://doi.org/10.1046/j.1365-8711.2000.03426.x.

[7] J. R. Hurley, C. A. Tout, and O. R. Pols. Evolution of binary stars and the effect of tides on binary populations. *Monthly Notices of the Royal Astronomical Society*, 329

(4):897–928, 02 2002. ISSN 0035-8711. doi: 10.1046/j.1365-8711.2002.05038.x. URL https://doi.org/10.1046/j.1365-8711.2002.05038.x.

[8] G. Iorio, G. Costa, M. Mapelli, M. Spera, G. J. Escobar, C. Sgalletta, A. A. Trani, E. Korb, F. Santoliquido, M. Dall'Amico, N. Gaspari, and A. Bressan. Compact object mergers: exploring uncertainties from stellar and binary evolution with SEVN. *arXiv e-prints*, art. arXiv:2211.11774, Nov. 2022. doi: 10.48550/arXiv.2211.11774.

[9] G. Iorio, M. Mapelli, G. Costa, M. Spera, G. J. Escobar, C. Sgalletta, A. A. Trani, E. Korb, F. Santoliquido, M. Dall'Amico, N. Gaspari, and A. Bressan. Compact object mergers: exploring uncertainties from stellar and binary evolution with sevn, 2023.

[10] M. Mapelli. Formation Channels of Single and Binary Stellar-Mass Black Holes. art. 16, 2021. doi: 10.1007/978-981-15-4702-7_16-1.

[11] P. C. Peters. Gravitational radiation and the motion of two point masses. *Phys. Rev.*, 136:B1224–B1232, Nov 1964. doi: 10.1103/PhysRev.136.B1224. URL https://link.aps.org/doi/10.1103/PhysRev.136.B1224.